

EL ORDENADOR PERSONAL



Nº 28

1984

450 pts.

ESTE NUMERO CONTIENE UNA SELECCION DE LOS MEJORES PROGRAMAS PUBLICADOS EN EL ORDENADOR PERSONAL

50 programas BASIC

juegos, enseñanza, aplicaciones, matemáticas, etc.

GRAN POSTER:

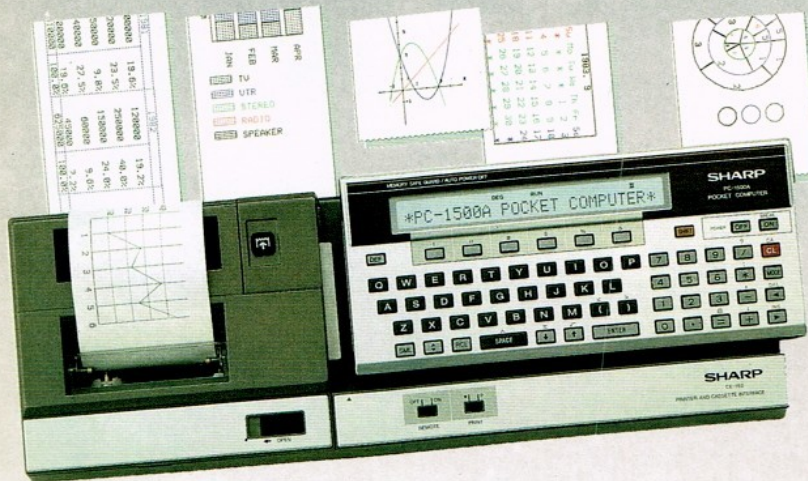
Con el cuadro comparativo de todos los Basic
para poder utilizar los programas en:

Apple, Atari 400, Atom, BBC, Commodore 64, DAI, Dragon 32, Epson HX 20, Laser, Lynx, MZ 80 A, New Brain, Oric 1, PET/CBM, Sord M-5, Spectravideo, Thomson T07, T1 99, TRS 80, Vic 20, ZX 81, ZX Spectrum

GENIOS DE BOLSILLO A TUS ORDENES...

Llévalos a cualquier lugar disfrutando de su potencia. Para toda actividad que realices: estudios, ingeniería, comercial, pasatiempos, análisis administrativos...

Sharp ofrece en tamaño billetera la combinación ideal de una gran facilidad operativa, disponibilidad de programas y superioridad de sistema.



PC-1500A

DISPLAY E IMPRESORA GRAFICA

Display gráfico 16KB ROM. Reloj interno. Basic extendido. Memoria continua. Teclas definibles. Impresora-plotter de 4 colores. Telecomunicación.

RAM 3.5KB (PC-1500) Ptas. 37.750,-
RAM 8.5KB (PC-1500 A) Ptas. 52.100,-

PC-1251

BASIC COMPUTER. ¡SOLO 115 GRAMOS!

Basic extendido. ROM 24KB. RAM 4.2KB.
Teclas reservables. Memoria continua.
Compatible con PC-1245/PC-1401 Ptas. 23.700,-
Opcional: Impresora-microcassette integrados.



PC-1245

COMPUTADORA ESTUDIANTIL

Basic extendido. ROM 25KB. RAM 2.2KB.
Teclas definibles. Memoria continua.
Compatible con PC-1251/PC-1401 Ptas. 16.700,-
Opcional: Impresora-microcassette integrados.

PC-1401

LA CIENTIFICA

Basic extendido. Memoria continua.
Teclas definibles y preprogramadas.
Gran número de funciones matemáticas y estadísticas.
Compatible con PC-1245/PC-1401 Ptas. 21.875,-
Opcional: Impresora-interface de cassette.



SHARP

Consíguelos en los distribuidores autorizados o en:

MECANIZACION DE OFICINAS, S.A.

BARCELONA-36: Av. Diagonal, 431-bis. Tel. 200 19 22

MADRID-3: Santa Engracia, 104. Tel. 441 32 11

BILBAO-12: Iparraguirre, 64. Tel. 432 00 88

VALENCIA-5: Ciscar, 45. Tel. 333 55 28

SEVILLA-1: San Eloy, 56. Tel. 21 50 85

ZARAGOZA-6: J. Pablo Bonet, 23. Tel. 27 41 99

HABLEMOS EN BASIC

Puede que os hayais tirado a menudo de los pelos delante de un programa BASIC de otro ordenador que os gustaría adaptar al vuestro. Ya seais adictos o iniciados, esta tabla que os presentamos será de una eficaz utilidad.

BASIC significa "de base", "lo más simple". BASIC es también el nombre de un lenguaje utilizado hoy en día por 6 millones de ordenadores en todo el mundo. ¿"El" lenguaje? Sí, en cierto modo, pero desgraciadamente, según el ordenador que se tenga, el BASIC presentará una o varias características específicas y por tanto conviene hablar de "los" lenguajes BASIC. Para ayudarle a adaptar los programas BASIC, hemos establecido una tabla comparativa que, para un centenar de instrucciones, da su "traducción" sobre dieciocho ordenadores personales corrientes.

La adaptación de programas BASIC de un OP a otro no plantea problemas serios en un 90% de las líneas de este. RUN, NEW, END, RETURN, etc... son prácticamente siempre las mismas.

Bien, pero ¿qué ocurre en el 10% restante? Este 10% encierra todas las variantes, todos los "falsos amigos", todas las instrucciones que cambian de sentido, todas las posibilidades de astucias particulares (sin duda las más complicadas), en pocas palabras, todas las "trampas" BASIC imaginables.

Tomemos por ejemplo un VIC-20 y un APPLE: Al ver la instrucción INPUT de lectura sobre disco del APPLE, podría pensarse que se trata de una entrada por el teclado del VIC-20, y modificar consecuentemente el programa. El ejemplo es simple pero GET, RND, CLR y otros CLS reservan desgraciadamente otras desagradables sorpresas.

Atención a las instrucciones que se refieren a la memoria de vuestro ordenador. Las órdenes PEEK, POKE, CALL y otras USR tendrán que ser práctica y sistemáticamente adaptadas. Los símbolos gráficos, el color y el sonido son ellos también muy específicos del ordenador utilizado y, aquí también, se hace preferible "adaptar" más que "traducir" de un ordenador a otro.

Por lo demás, os vais a encontrar en las páginas que siguen un total de cien instrucciones BASIC Microsoft clasifi-

cadas por orden alfabético. De esta forma, la tabla podrá utilizarse no sólo como diccionario de instrucción BASIC, sino también como tabla de traducción de diferentes comandos de un material a otro.

Es de notar que, por razones evidentes de espacio (teniendo en cuenta el importante número de instrucciones, siendo muchas de ellas específicas de un material), esta tabla no es todo lo exhaustiva que sería deseable. Como es más descriptiva que explicativa, conviene utilizarla en paralelo con el manual de utilización de los materiales considerados.

Tabla de correspondencia			
<i>En esta tabla se reseñan las instrucciones específicas de un ordenador y su correspondiente instrucción del BASIC Microsoft.</i>			
Apple DEL GET RENAME NOTRACE TRACE	DELETE INKEYS NAME TROFF TRON	LN TRACE OFF TRACE ON	LOG TROFF TRON
		CLR MEM GET LN BYE	MZ 80 A CLEAR FRE INKEYS LOG SYSTEM
Atari 400 CLR CLOG BYE GRAPHICS	CLEAR CLG SYSTEM TEXT	New Brain COLOR FRE OUT SCRN	
		COL FREE PUT PEN (6)	COLOR FRE OUT SCRN
Atom CH LINK \$	ASC CALL, USR CHRS	Oric 1 GET, KEYS LN CURSET	
		Sord M-5 ASC DELETE LOG TRON TROFF	
BBC LOG LN BPUT RENUMBER TRACE OFF TRACE ON	CLG LOG OUT RENUM TROFF TRON	Spectravideo PSET RND	
		TI 99 NUM OLD CALL CLEAR CON RES UNTRACE TRACE	
Commodore 64 CLR	CLEAR	TRS 80 modelo 1 CINT SET SPACES	
		Vic 20 SYS CLR GET	
Dai ACOS ASIN LOGT DOT	ACS ASN CLG PLOT	ZX 81 LN LOG	
		ZX Spectrum LN LOG	
Dragon 32 EXEL DEL MEM PSET	CALL DELETE FRE PLOT	Apple DEL GET RENAME NOTRACE TRACE	
		Atari 400 CLR CLOG BYE GRAPHICS	
Epson EXEC PSET	CALL PLOT	Atom CH LINK \$	
		BBC LOG LN BPUT RENUMBER TRACE OFF TRACE ON	
Laser SET POINT	PLOT SCRN	Commodore 64 CLR	
		Dai ACOS ASIN LOGT DOT	
Lynx ARCOS ARCSIN ARCTAN LOG DEL ROUND GETS, KEYS	ACS ASN ATN CLG DELETE FIX INKEYS	Dragon 32 EXEL DEL MEM PSET	
		Epson EXEC PSET	

SUMARIO

Director:

Javier San Román.

Director Adjunto:

S.M. Peyrou.

REDACCION:

Coordinador de Redacción:

S.M. Peyrou.

Director Técnico:

J. Antonio Deza.

Jefe de Redacción:

José Luis Sanabria.

Secretaria de Redacción:

Yolanda Hernández.

Diseño Gráfico:

Carlos Gorrindo.

Composición:

M^a Dolores Raboso.

Montaje:

Vicente Hernández.

Fotografía:

Barahona.

Colaboradores: S. Almeida - Antonio Bellido - Iñaki Cabrera - Alfonso Cachinero Sánchez - Víctor Manuel Delgado - José Antonio Deza Navarro - Víctor Manuel Díaz - Pedro Díaz Cuadra - Jaime Díez Medrano - José María Espinosa Fernández - Fabio Gil Miguel - Santiago González Ascensión - Félix Gutiérrez Fernández - Jesús Gutiérrez Peregrina - Ian Hinton - Gerardo Izquierdo Cadalso - Miguel Ángel Lerma Usero - José Antonio Mañas Valle - Valentín Martín González - José Francisco Martínez Antonioni - Justo Maurín - Antonio Miguel Morales Elbar - Manuel Otero Raña - Alberto Requena Rodríguez - José María Rodríguez Prolongo - Francisco Romero - Víctor Manuel Sevilla - Ricardo Trigo Calonge - José María Vicens Gómez - José María Vidal Lacasa.

PUBLICIDAD - VENTAS Y ADMINISTRACION:

Director de Publicidad:

Santiago Mondet.

Asistido por:

Yolanda Hernández.

Administración:

Mariano Alonso Sánchez.

Suscripciones:

Lucía Pérez.

REDACCION - PUBLICIDAD ADMINISTRACION:

Para España y Extranjero:

Calle Ferraz, 11, 3^o

MADRID-8

Tel.: (91) 247 30 00 - 241 34 00

Imprenta:

Pentacrom, S.L.

Hachero, 4 - Madrid.

Distribuye:

SGEL

Avda. Valdeparra, S/N

Alcobendas (Madrid)

El Ordenador Personal expresa sus opiniones sólo en los artículos sin firma. El resto de los conceptos tratados responde exclusivamente a la opinión y responsabilidad de sus autores y colaboradores.

La presente publicación ha sido confeccionada en parte, con material del Ordinateur Individuel con cuya editorial se ha suscrito un contrato temporal de colaboración.

1 Hablemos en BASIC (explicación del poster)

5 Editorial.

APLICACIONES

11 ZX-81: Rutina en código máquina para proteger programas en Basic.

15 Geografía Espacial. PC-1500.

19 TIC-TAC en VIC y en ORIC.

26 Traductor-Monitor/Intérprete para MZ-80 B.

28 Gestión Familiar. Cuando el Ordenador administra su tesorería.

36 Gráficos en el Osborne.

38 Informática de bolsillo. Cambie de base cuando lo necesite con este programa para Calculadoras HP.

50 Códigos de Barras, IMPRESORA C-ITOH 8510.

65 Si está perdido sitúese con un mapa y una calculadora de bolsillo.

68 BASIÑOL. BASIC en español.

85 Realizad nuevos caracteres en vuestro VIC-20.

102 Gestión de Ficheros. Mantenimiento de una Biblioteca.

108 BIG PATTERN (Gran silueta).

119 Choque Elástico.

125 Supervisión de cuentas corrientes con un SHARP-PC-1211.

139 Cálculo de velocidad de Perfusión de Fármacos Vasoactivos.

145 El Ordenador y la HI-FI.

164 Medidor de velocidad de cassette para ATOM.

166 Programa para alta resolución en el ZX-81.

JUEGOS

7 El Laberinto de CANDY.

24 QUILOPUS.

33 El juego de NEISCAT.

38 Juego de las Siete y Media para la CASIO FX-702 P.

45 La Huida con obstáculos.

47 Los invasores han vuelto, yo los he encontrado.

53 Gran Premio de PENCHES.

55 Programas de 1K. para el ZX-81.

59 El juego del Radar, para SHARP PC-1211.

74 OP. DEFENDER.

79 Invasión Antártica.

83 ¿Conseguirá el ZX-81 salvar a los naufragos?

89 El Juego del Globo.

99 Pánico en el fondo del mar.

105 Rally de Montecarlo, para PC-1500.

112 A hacer chuletas, para ZX-81.

123 Avión Espía, para la TI-57.

127 Máquina Tragaperras OP-FRUIT (Spectrum).

133 MINIGOLF.

143 Programas de 1K. para ZX-81.

150 ¿Espíritu estás aquí?: Los Fantasmas del Comodore. 1ª Parte.

153 ¿Espíritu estás aquí?: Los Fantasmas del Comodore. 2ª Parte.

157 ¿Espíritu estás aquí?: Los Fantasmas del Comodore. 3ª Parte.

159 ¿Espíritu estás aquí?: Los Fantasmas del Comodore. 4ª Parte.

171 Laberinto para MZ-80 B.

ENSEÑANZA

93 Enseñe al Ordenador Geografía.

135 Aprendizaje del BASIC en un Instituto de Bachillerato.

141 Las Calculadoras Programables también sirven para aprender.

MATEMATICAS

23 Integración de métodos de SIMPSON con el ZX-81.

62 Producto de matrices.

114 División de Polinomios.

Si Ud. necesita otro poster
solicítelo
Precio: 300 pts.

NÚMERO ESPECIAL

1ª REVISTA DE ORDENADORES PERSONALES

EL ORDENADOR PERSONAL

Nº 28 1984 450 pts.

ESTE NÚMERO CONTIENE UNA SELECCIÓN DE LOS MEJORES PROGRAMAS PUBLICADOS EN EL ORDENADOR PERSONAL

50 programas BASIC

juegos, enseñanza, aplicaciones, matemáticas, etc.

GRAN POSTER:

Con el cuadro comparativo de todos los Basic para poder utilizar los programas en:

Apple, Atari 400, Atom, BBC, Commodore 64, DAI, Dragon 32, Epson HX 20, Laser, Lynx, MZ 80 A, New Brain, Orc 1, PET/CSII, Sord M 5, Spectravideo, Thomson TD7, TI 99, TRS 80, Vic 20, ZX 81, ZX Spectrum

Este número especial ha sido realizado a partir de artículos ya publicados en los números anteriores del Ordenador Personal.

DEFINITIVAMENTE, sepa dar solución a esas dificultades que le impiden un correcto funcionamiento de su empresa.

PORQUE... ESTO ES LO QUE ANDABA USTED BUSCANDO

Un sistema eficaz que la mejore, solucionando esas eternas dificultades: en la facturación, las nóminas, el control presupuestario, la información, los mailings, el stock de producción, etc. y de hacer por tanto una más perfecta gestión en la actividad que usted realiza: Comercialización, Investigación, Enseñanza, etc.

Y ESTO ES MAYBE

La solución, una empresa con gran experiencia en la comercialización de los mejores miniordenadores del mercado, (casi tan rentables y eficaces como cualquier gran ordenador y notablemente más económicos).

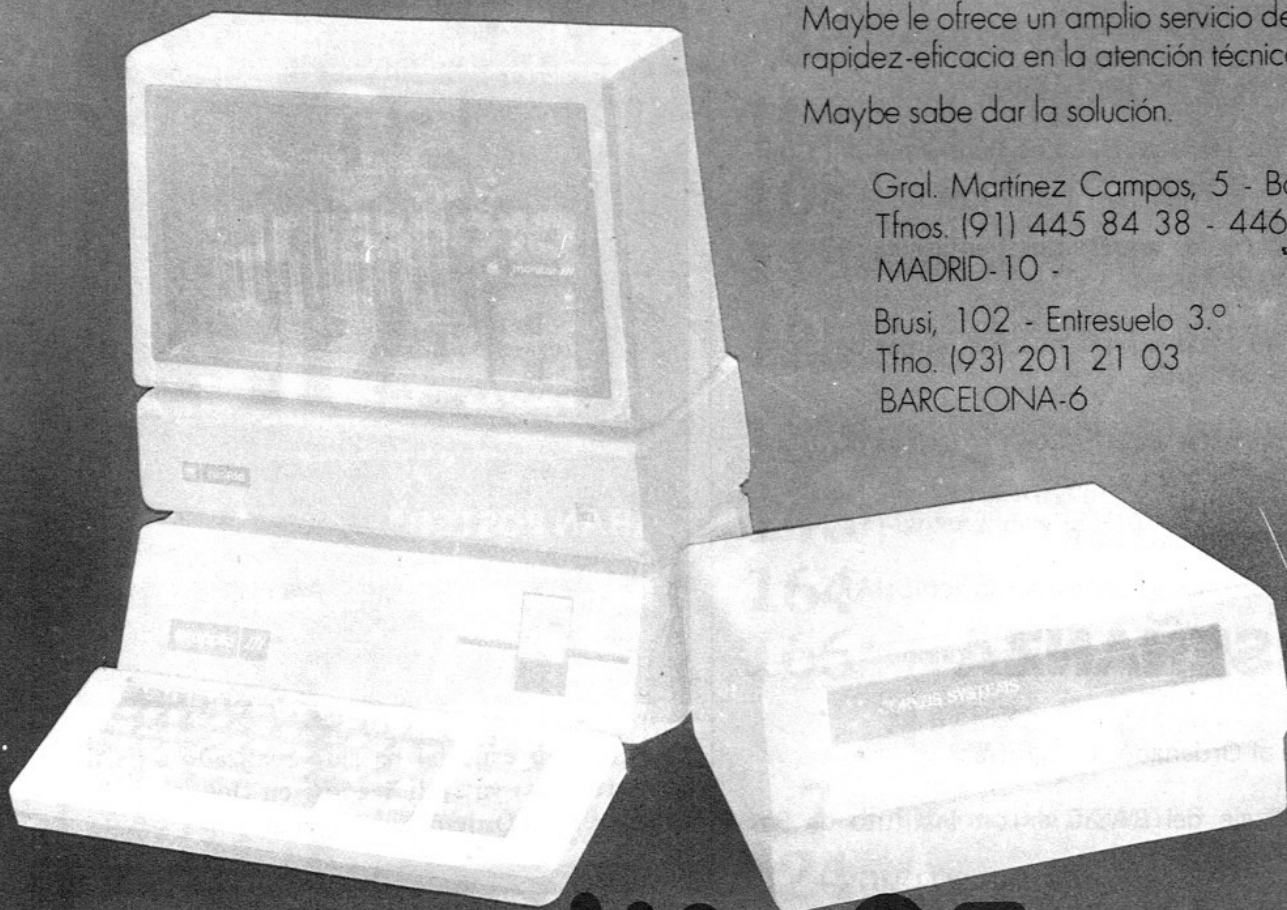
Maybe le garantiza un estudio en particular de su problema asesorándole en la compra del miniordenador más idóneo.

Maybe le ofrece un amplio servicio de Software y rapidez-eficacia en la atención técnica postventa.

Maybe sabe dar la solución.

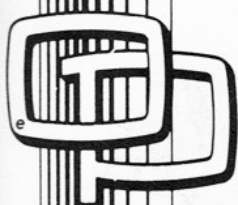
Gral. Martínez Campos, 5 - Bajo izqda.
Tfnos. (91) 445 84 38 - 446 60 18
MADRID-10 -

Brusi, 102 - Entresuelo 3.º
Tfno. (93) 201 21 03
BARCELONA-6



MAYBE

Electrónica y Servicios.



Editorial

Muchos han sido los programas publicados por el ORDENADOR PERSONAL en su larga etapa recorrida desde Febrero de 1982. Numerosos también son los lectores que nos piden una recopilación de los más interesantes.

Ha sido difícil la tarea de seleccionar estos artículos dada la abundancia de equipos y preferencias de nuestros lectores. Incluso hemos tenido que incluir algún programa para pequeños equipos que no funcionan con BASIC, y ello con el único fin de satisfacer al mayor número posible de usuarios —entre los cuales están, no lo olvidamos, los usuarios de calculadoras programables—.

En cuanto a la temática de los programas seleccionados, hemos cubierto principalmente cuatro grandes áreas: Juegos, Aplicaciones, Enseñanza y Matemáticas.

No obstante, dada la diversidad de sistemas con BASICs más o menos adaptados a un caso particular, hemos juzgado imprescindible publicar una tabla de traducción que recoja 100 instrucciones BASIC Microsoft y sus correspondientes traducciones para 22 ordenadores personales de uso extendido. Esta tabla, tamaño Poster, se incluye junto con el presente número. Con ello esperamos agilizar su consulta y al mismo tiempo dar un toque de originalidad.

Esperamos la aparición de un BASIC realmente estándar, disponible en todo ordenador personal.

El Editor

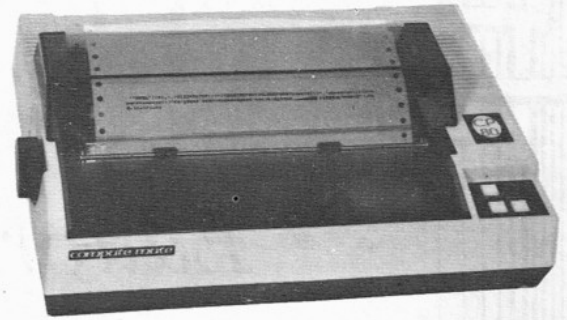
Equipos informáticos

DATALEC



DATALEC Monitor monocromo para visualización de datos.

El monitor DATALEC, con su pantalla de fósforo verde P-31 de 12 pulgadas, es la pantalla de visualización ideal para presentación de datos y gráficos en alta resolución. La carcasa es de ABS, resistente y fácil de limpiar, con un diseño estético muy elaborado, acorde al uso a que va destinado para conjuntar con cualquier ordenador de sobremesa. Dispone de mandos de luz y contraste, así como ajustes externos de entrada video, frecuencia vertical y altura. En pantallas de visualización de datos, el nombre es DATALEC.



SHINWA CP80 F/T

SHINWA Impresora matricial 80 columnas con set de caracteres españoles, totalmente compatible.

SHINWA CP80 F/T es la nueva impresora. Con tecnología actual y precio competitivo, ofrece las dos características que hoy día hay que exigir a una buena impresora: fiabilidad y calidad de impresión. Pero la SHINWA CP80 F/T no se queda ahí: ofrece una resolución de 640 puntos por línea, juego de caracteres españoles y una gran variedad de posibilidades en la impresión de textos: normal, comprimido, doble ancho, super índices subíndices reducidos, etc. La impresora se suministra con interface tipo CENTRONICS. Opcionalmente, se puede conectar un interface RS-232.

UNITRON

Su computador personal compatible

COMPUTADORAS PERSONALES, DE GESTION Y APRENDIZAJE



Ordenadores personales, de gestión y para aprendizaje. Dos marcas con prestigio que cubren todas las necesidades, desde el ordenador para aprender a programar hasta el ordenador que resuelve los problemas de la pequeña empresa (contabilidad, facturación, clientes), incluyendo unidades de disco flexible y tarjetas de expansión para adaptar el ordenador a sus necesidades.

ZX Spectrum

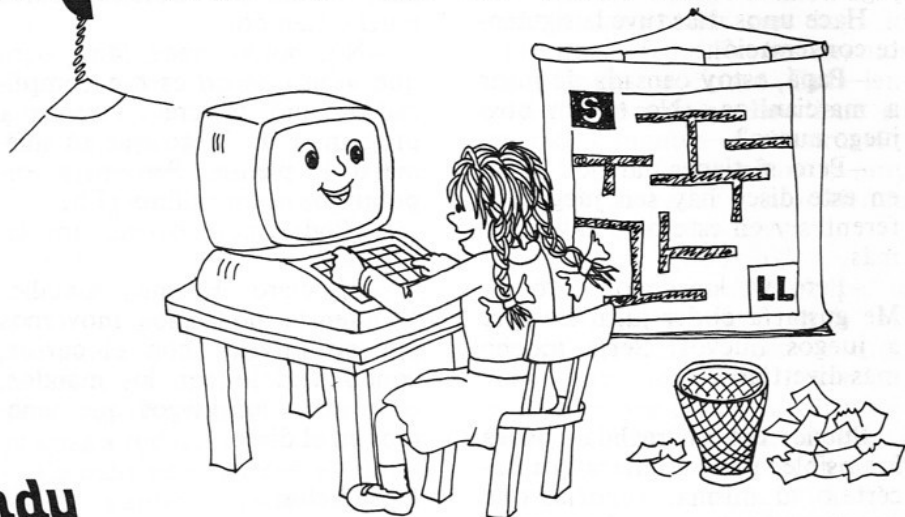


PROGRAMAS PARA ZX-SPECTRUM

Programas en cassette para su ZX-SPECTRUM. Los mejores programas con traducción al español de su manejo, a precios realmente competitivos.

Importador:
SITELSA, Equipos Electrónicos Avanzados
C/ Montaner, 44 - BARCELONA 11
TLX 54218 SITE

EL LABORATORIO DE CANDY



S. Almeida y Candy

La introducción de los niños en el mundo de los ordenadores, no es ninguna fantasía.

Hoy día está al alcance de nuestra mano. La instrucción puede combinarse perfectamente con la diversión y obtener resultados espectaculares.

Este artículo pretende ser una invitación para abrir esta puerta.

Introducción

Todos los padres nos encontramos, al inicio del verano, con el problema de tener que dejar a los hijos en casa, mientras no tenemos vacaciones, puesto que las de ellos son siempre mucho más largas.

Durante este período, el pequeño Apple II es un buen compañero de juegos, y proporciona muchas horas de aventuras, viajando por el espacio, haciendo test, persiguiendo extraños monstruos, rescatando princesas, etc.

Pero llega un momento en que el niño quiere hacer sus propios

juegos, saber cómo funcionan las cosas, acercarse a las "tripas" del juguete. Es importante dar satisfacción a este instinto creador y dejar que la fantasía pueda volcarse, aunque sea en muy pequeña medida. Con ello, sin duda, estamos formando un cerebro amplio, capaz de entender el futuro al que estamos abocados.

Este artículo es una corta historia de esta inquietud. Quizás el lector se encuentre con un problema similar, y le pueda servir como guía.

El problema

Candy tiene 13 años. Le gusta mucho jugar con el ordenador personal que tenemos en la mesa del despacho, cuando está

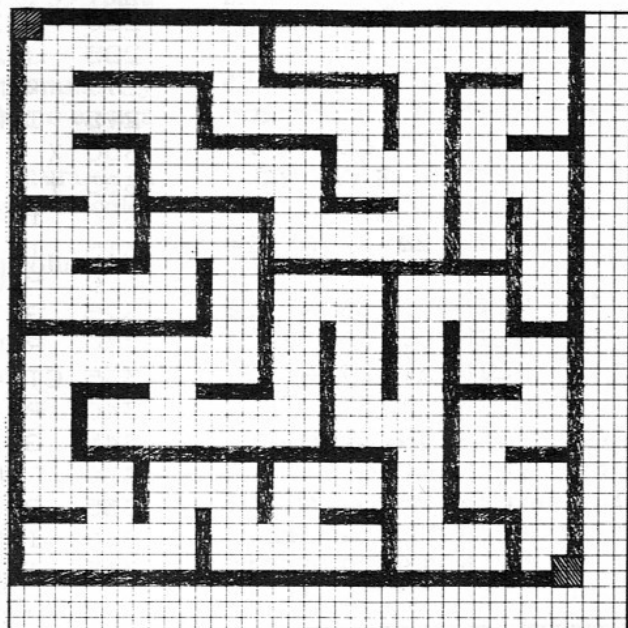
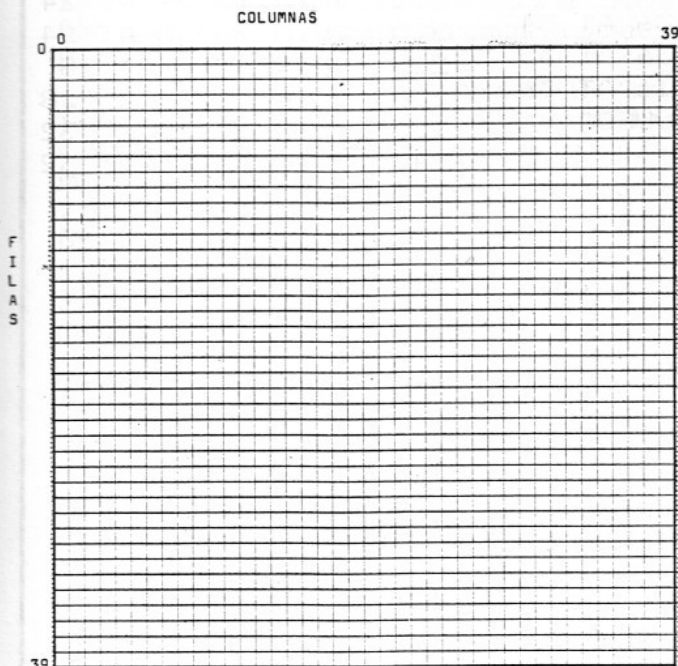


Figura 1

cansada de estudiar y no es momento de salir al parque a jugar.

Hace unos días tuve la siguiente conversación:

—Papá, estoy cansada de jugar a marcianitos. ¿No tienen otro juego nuevo?.

—Pero si tienes muchos. Mira, en este disco hay seis juegos diferentes y en este otro hay cinco más.

—Pero ya los conozco todos. Me gustaría poder jugar siempre a juegos nuevos. Sería mucho más divertido.

Bueno, eso en realidad no es imposible, pero tendrías que hacértelo tú misma. Tendrías que programarte tus propios juegos y así, cada vez que quieras, te los puedes cambiar.

— ¡Huy!. Pero eso será muy complicado. Tendría que estudiar mucho y también tardaría mucho tiempo.

—No, no lo creas. Mira, para que veas que no es tan complicado como te crees, vamos a programar un juego que tú misma te lo pienses. Pero para empezar que sea sencillito ¿Eh?.

—¿Podríamos hacer un laberinto?.

—Sí, claro. Es muy sencillo. Y además podremos movernos por su interior con el cursor, conduciéndolo con los mandos, como con los juegos que tenemos en el disco.

La solución

Ya decididos a programar nuestro nuevo juego, tomamos un

```

2 HOME
5 GR
17 COLOR= 15
25 HLIN 0,36 AT 0
30 HLIN 0,36 AT 36
35 VLIN 0,36 AT 0
40 VLIN 0,36 AT 36
50 HLIN 4,12 AT 4
55 HLIN 16,24 AT 4
60 HLIN 28,32 AT 4
65 HLIN 4,8 AT 8
70 HLIN 12,20 AT 8
75 HLIN 32,36 AT 8
80 HLIN 0,4 AT 12
85 HLIN 8,16 AT 12
90 HLIN 20,24 AT 12
95 HLIN 4,8 AT 16
100 HLIN 16,32 AT 16
105 HLIN 0,12 AT 20
110 HLIN 32,36 AT 20
115 HLIN 4,8 AT 24
120 HLIN 12,16 AT 24
125 HLIN 28,32 AT 24
130 HLIN 4,24 AT 28
135 HLIN 32,36 AT 28
140 HLIN 0,4 AT 32
145 HLIN 20,24 AT 32
150 HLIN 28,32 AT 32
155 VLIN 24,28 AT 4
160 VLIN 8,16 AT 8
165 VLIN 28,32 AT 8
170 VLIN 4,8 AT 12
175 VLIN 16,20 AT 12
180 VLIN 32,36 AT 12
185 VLIN 0,4 AT 16
190 VLIN 12,24 AT 16
195 VLIN 28,32 AT 16
200 VLIN 8,12 AT 20
205 VLIN 20,28 AT 20
210 VLIN 4,8 AT 24
215 VLIN 16,24 AT 24
220 VLIN 28,36 AT 24
225 VLIN 4,16 AT 28
230 VLIN 20,32 AT 28
235 VLIN 12,20 AT 32
240 VLIN 32,36 AT 32
245 REM SALIDA/LLEGADA
250 COLOR= 11
255 HLIN 0,2 AT 0
260 HLIN 0,2 AT 1
265 HLIN 0,2 AT 2
270 COLOR= 12
275 HLIN 34,36 AT 34
280 HLIN 34,36 AT 35
285 HLIN 34,36 AT 36
290 REM BUSQUEDA
295 COLOR= 9
300 X = PDL (0) / 6
305 Y = PDL (1) / 6
310 IF SCRN( X,Y) = 15 THEN GOTO 335
315 IF SCRN( X,Y) = 12 THEN GOTO 355
320 PLOT X,Y
325 GOTO 300
330 REM FALLO
335 CT = CT + 1
340 PRINT "FALLOS:";CT
345 GOTO 300
350 REM LLEGADA
355 PRINT "FIN DE LA PARTIDA"
360 INPUT R
365 TEXT
370 END

```

```

0 REM PREPARACION
2 HOME
5 GR
10 CT = 0
15 PRINT "FALLOS:";CT
17 COLOR= 15
20 REM CUADRO
25 HLIN 0,36 AT 0
30 HLIN 0,36 AT 36
35 VLIN 0,36 AT 0
40 VLIN 0,36 AT 36
45 REM ENTRAMADO
50 HLIN 4,12 AT 4
55 HLIN 16,24 AT 4
60 HLIN 28,32 AT 4
65 HLIN 4,8 AT 8
70 HLIN 12,20 AT 8
75 HLIN 32,36 AT 8
80 HLIN 0,4 AT 12
85 HLIN 8,16 AT 12
90 HLIN 20,24 AT 12
95 HLIN 4,8 AT 16
100 HLIN 16,32 AT 16
105 HLIN 0,12 AT 20
110 HLIN 32,36 AT 20
115 HLIN 4,8 AT 24
120 HLIN 12,16 AT 24
125 HLIN 28,32 AT 24
130 HLIN 4,24 AT 28
135 HLIN 32,36 AT 28
140 HLIN 0,4 AT 32
145 HLIN 20,24 AT 32
150 HLIN 28,32 AT 32
155 VLIN 24,28 AT 4
160 VLIN 8,16 AT 8
165 VLIN 28,32 AT 8
170 VLIN 4,8 AT 12
175 VLIN 16,20 AT 12
180 VLIN 32,36 AT 12

```

```

185 VLIN 0,4 AT 16
190 VLIN 12,24 AT 16
195 VLIN 28,32 AT 16
200 VLIN 8,12 AT 20
205 VLIN 20,28 AT 20
210 VLIN 4,8 AT 24
215 VLIN 16,24 AT 24
220 VLIN 28,36 AT 24
225 VLIN 4,16 AT 28
230 VLIN 20,32 AT 28
235 VLIN 12,20 AT 32
240 VLIN 32,36 AT 32
245 REM SALIDA/LLEGADA
250 COLOR= 11
255 HLIN 0,2 AT 0
260 HLIN 0,2 AT 1
265 HLIN 0,2 AT 2
270 COLOR= 12
275 HLIN 34,36 AT 34
280 HLIN 34,36 AT 35
285 HLIN 34,36 AT 36
290 REM BUSQUEDA
295 COLOR= 9
300 X = PDL (0) / 6
305 Y = PDL (1) / 6
310 IF SCRN( X,Y) = 15 THEN GOTO 335
315 IF SCRN( X,Y) = 12 THEN GOTO 355
320 PLOT X,Y
325 GOTO 300
330 REM FALLO
335 CT = CT + 1
340 PRINT "FALLOS:";CT
345 GOTO 300
350 REM LLEGADA
355 PRINT "FIN DE LA PARTIDA"
360 INPUT R
365 TEXT
370 END

```


lápiz, y un papel cuadriculado y nos sentamos en una mesa, dispuestos a esta nueva aventura. Nuestra conversación continuó así:

—Mira, Candy, para hacer el laberinto en el ordenador, primero tenemos que hacer un diseño sobre una hoja de papel. En este libro dice que para hacer dibujos de baja resolución, que son los más sencillos, tenemos que acomodarnos a un cuadriculado de 40 x 40 casillas.

—¿Así? (ver figura 1).

—Exacto. Ahora vamos a dibujar tu laberinto. Tienes que tener cuidado de dejar suficiente espacio para que el cursor se mueva cómodamente por su interior.

Por ejemplo, que el pasillo sea de 3 cuadraditos de ancho.

—¿Qué te parece? (ver figura 2).

—¡Estupendo!. Ahora ya tenemos nuestro diseño. Sólo falta programarlo.

El programa

Nos sentamos delante del teclado de nuestro ordenador, introducimos el disco del sistema en el lector de discos, le damos al interruptor del monitor de vídeo y al interruptor del teclado. Se enciende la luz de potencia (Power), y el disco comienza a rodar.

Cuando nos aparece el cursor en la pantalla, estamos listos

para empezar a darle instrucciones al ordenador.

—Bueno, Candy. Vamos a decirle al ordenador que haga lo que queremos. Se lo diremos dándole instrucciones. Cada lenguaje de programación tiene su propio conjunto de instrucciones. Nosotros necesitamos muy pocas. Pero ya las iremos viendo a medida que las utilizemos.

—Y ahora ¿qué hago?

—Primero dejaremos limpia la pantalla, eso se dice con la instrucción HOME, luego le diremos que queremos hacer un gráfico, eso se dice con la instrucción GR. Después que las líneas sean blancas, se dice con COLOR = 15.

—¿Así?.

—De acuerdo. Ahora sólo falta hacer el dibujo y añadir algunos detalles para mover el cursor.

—¿Y cómo se hace el dibujo?

—Muy fácil, las líneas verticales las indicaremos con VLIN a,b AT c, que traza una línea vertical en la columna "c" desde la fila "a" a la "b". Igualmente las líneas horizontales las haremos con HLIN a, b AT c, que permite trazar una horizontal sobre la fila "c" desde la columna "a" a la "b".

—¿Qué te parece? (ver listado 1).

—¡Muy bien!. Ahora añadiré yo algunos detalles para contar los fallos y mover el cursor. Ves, así. Ya tenemos el programa listo (ver listado 2). Ahora damos la instrucción SAVE, y ya lo tenemos guardado en el disco. Cuando quieras, podemos utilizarlo, igual que los otros de marcianitos.

Conclusión

Este sólo ha sido el principio, ahora ya está interesada en los dibujos de alta resolución, en cómo dejar fijos los paisajes del fondo moviendo figuras independientemente de éste. Es muy divertido crear pequeños monstruos moviéndose por paisajes extravagantes, conducidos por nuestros mandos.

Jugando, nos podemos introducir en una nueva técnica, que está llamada a ser parte muy importante del futuro inmediato de nuestro mundo.

S. Almeida

INSTRUCCIONES PARA GRAFICOS DE BAJA RESOLUCION

Cuando queremos dibujar sobre la pantalla de nuestro ordenador personal, tenemos que hacer tres cosas diferentes:

1. Indicar que abandonamos el modo texto y pasamos al modo gráfico, mediante la instrucción GR.
2. Indicar el color que usaremos en las próximas instrucciones de ejecución mediante la instrucción COLOR=a, siendo a un número de los indicados en la tabla de colores.
3. Realizar el dibujo con las instrucciones que se indican más adelante.

Adicionalmente, si nuestro dibujo se complica mucho, puede ser necesario saber la situación en que se encuentra una determinada casilla, para pasarla por alto o para cambiarla de color. Para este caso tenemos otro tipo de instrucción:

4. Interrogación del estado, es decir, obtención del número correspondiente al color que tiene en ese momento. Para ello se utiliza la instrucción SCRN (X, Y), donde X e Y representan el número de la columna y de la fila, respectivamente, que se desea consultar.

Es muy frecuente utilizar la letra X para referirse a la columna que deseamos referenciar, y la letra Y de forma similar para la fila que deseamos referenciar. De esta forma, si el par (X,Y) tiene los valores X=2 e Y=5, nos estamos refiriendo a la tercera columna y a la sexta fila, puesto que la numeración empieza desde el 0 hasta el 39 (para las 40 filas y columnas).

Para realizar el dibujo disponemos de tres instrucciones:

1. PLOT X,Y: Colores el cuadro correspondiente a la columna X y a la fila Y, con el color cuyo valor se ha indicado por último en una instrucción COLOR=a.
2. HLIN X1, X2 AT Y: Traza una línea horizontal a la altura de la fila Y, desde la columna X1 hasta la columna X2, según el color indicado.
3. VLIN Y1, Y2 AT X: Traza una línea vertical sobre la columna X desde la fila Y1 a la Y2, según el color indicado.

TABLA DE COLORES

0 —negro	8 —marrón
1 —magenta	9 —naranja
2 —azul oscuro	10 —gris
3 —púrpura	11 —rosa
4 —verde oscuro	12 —verde
5 —gris	13 —amarillo
6 —azul medio	14 —azul
7 —azul claro	15 —blanco



GEMINI 10X : 80 COLUMNAS, 120 cps.
GEMINI 15X : 132 COLUMNAS, 120 cps.



Delta 10 : 80 columnas, 160 cps.
Delta 15 : 132 columnas, 160 cps.



IMPRESORAS **star**



Radix 15 : 80 columnas, 200-38 cps.
Radix 15 : 132 columnas, 200-38 cps.



Powertype : 110 - 132 - 165 columnas, 18 cps.

De venta en establecimientos especializados.

IMPORTADO POR

 **SCS**
COMPONENTES ELECTRONICOS, S. A

Gran Via de les Corts Catalanes, 682, Barcelona-10
Teléfonos 318 85 33 - 318 89 12
Telex 50204 SCS E

ZX81: Rutina en código máquina para proteger programas en Basic



Muchos autores desearían disponer de un sistema de protección que impidiera que los programas que diseñan sean fáciles de copiar por otros usuarios. Aquí presentamos una sencilla rutina en CM, fácil de incorporar a cualquier programa escrito en BASIC, y destinada a protegerlo frente a «espías» y copias incontroladas.

La protección de programas

Un programa se dice «protegido» cuando dispone de un mecanismo destinado a impedir, o dificultar seriamente, el acceso de los usuarios a su listado y a la posibilidad de copiarlos y grabarlos. La protección de programas tiene como finalidad entorpecer las copias incontroladas y el plagio.

Probablemente no existe un procedimiento perfecto de protec-

ción. En el ZX81 nada impide que un programa grabado en cinta se pueda reproducir de cassette a cassette si se emplean aparatos que distorsionen poco la señal; pero la copia nunca tendrá la calidad del original, y esto impone un límite natural a este procedimiento de reproducción.

En cuanto al acceso al listado, existe un procedimiento bastante bueno que lo previene: diseñar el programa enteramente en CM, grabarlo de modo que se autoejecute (mediante una instrucción «SAVE» en medio del programa) y

evitar que devuelva el control a BASIC. Como el programa no se podrá detener, tampoco se podrá listar ni grabar. Incluso si el programa se detiene, pocos usuarios entienden el CM, y esto es una buena garantía. Pero al mismo tiempo es un inconveniente, porque pocos usuarios podrán proteger sus programas a base de diseñarlos exclusivamente en CM. Además, este procedimiento de programación resulta a veces muy incómodo y farragoso a pesar de sus ventajas, especialmente si hace falta manejar largos mensajes en la pantalla, u operaciones en coma flotante.

Debe señalarse aquí un curioso truco que impide la autoejecución de un programa, y que ya fue publicado en ORDENADOR PERSONAL (nº 14). Consiste en cargar mediante la secuencia de comandos: FAST, NEWLINE, RAND USR 837, NEWLINE, RAND USR 837, NEWLINE, (y se pone en

```

1 REM 12345678901234567890123
45678901234567890123456789012345
67890123456789012345678901234567
8901234567890
20 LET A$="DD22BB40DD218B40C9E
1D1C1F122BD4021AB40E52ABD40F5C5D
5E5B6D3FD2A0C40CBFCDD2ABB40DDE9"
22 LET A$=A$+"CD8240F5083C0083A
00403CC40000F1C900000000CDE702FD
CB3B562A0C4036C7C92A0C40367600FD
CB3BF6CD0702C38240"
30 FOR X=1 TO LEN A$ STEP 2
40 LET A=16*(CODE A$(X)-26)+CO
DE A$(X+1)-26
50 POKE (16514+(X-1)/2),A
60 NEXT X
70 STOP
9980 REM GRABACION
9990 SAVE "PROTECTOR"
```

```

1 REM <>60RND<>50RNDTAN LPRIN
T SGN AT LET 60RND50RND FAST E0R
ND PRINT VAL STR$ FAST 0PEEK CLE
AR E0RNDACS UNPLOT <>E0RND<> DIM
LN L0RND PRINT 0000 RNDUCODE L
ET TAN LN SCROLL CLEAR ACS
V0E0RND00SIN TAN E0RND0
253 PLOT LN 0000L0RND1234567890
9980 REM GRABACION
9990 SAVE "PROTECTOR"
```

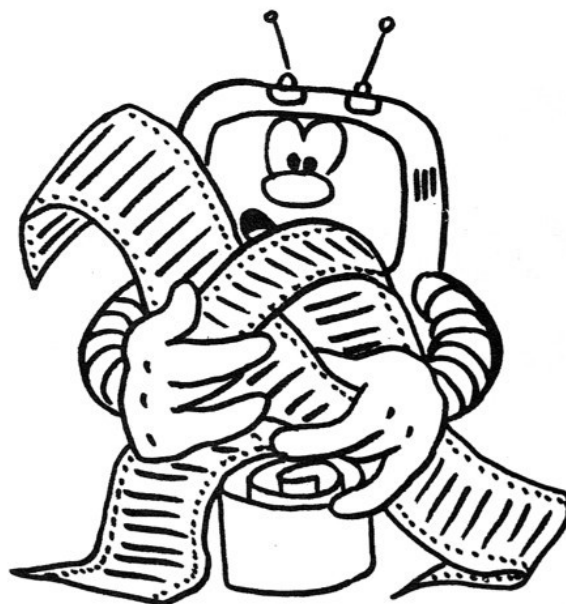
```

1 REM <>60RND<>50RNDTAN LPRIN
T SGN AT LET 60RND50RND FAST E0R
ND PRINT VAL STR$ FAST 0PEEK CLE
AR E0RNDACS UNPLOT <>E0RND<> DIM
LN L0RND PRINT 0000 RNDUCODE L
ET TAN LN SCROLL CLEAR ACS
V0E0RND00SIN TAN E0RND0
253 PLOT LN 0000L0RND1234567890
9980 REM GRABACION
9985 IF USR 16575 THEN REM
9990 SAVE "PROTECTOR"
9995 IF USR 16588 THEN REM
```

```

1 REM <>6RND<>5RNDTAN LPRIN
T SGN AT LET 6RND5RND FAST E
ND PRINT VAL STR$ FAST PEEK CLE
AR EARNDCS UNPLOT <>EARNDC<> DIM
LN RND PRINT WU RNDUCODE L
ET TAN ?INKEY$LN SCROLL CLEA
R ACS VEARNDCSIN TAN EARNDC
253 PLOT LN ?RND1234567890
100 REM CLAVE DE
    DESPRIVATIZACION
110 INPUT M$
120 IF M$="ABO" THEN GOTO 9000
130 GOTO 100
9000 REM DETENCION DEL PROGRAMA
9010 FAST
9020 SLOW
9030 STOP
9980 REM GRABACION
9985 IF USR 16575 THEN REM
9990 SAVE "PROTECTOR"
9995 IF USR 16588 THEN REM
9998 GOTO 100

```



marcha el cassette). Cualquier procedimiento de protección deberá tener previsto este truco.

El principal problema en la protección de un programa en BASIC consiste en que éste siempre puede detenerse mediante BREAK o STOP, o bien forzando un error. No vale de nada, por tanto, diseñar el programa de modo que se autoejecute al cargar y no se detenga nunca por sí solo; ya que el usuario podrá detenerlo cuando él quisiera. No obstante, cualquier detención forzará un error, cuyo código aparece al pie de la pantalla y que indica la causa de la detención. Por ejemplo, BREAK fuerza un error de código «D», y STOP uno de código «9» (o «D»). Esto puede ser aprovechado por una rutina en CM, que se encargará de eje-

cutar un borrado automático de la memoria al detectar un error de código distinto de cero.

Para que el mecanismo de protección que aquí se propone sea eficaz, el programa en BASIC deberá cumplir las siguientes condiciones:

1. Debe autoejecutarse al cargarlo.
2. Nunca debe detenerse por sí solo.
3. Debe ser imposible detenerlo sin forzar un error de código no nulo.

Respecto a la última condición, debe recordarse que el programa se detendrá dando un código de error nulo si alcanza la última línea del programa (a no ser que ésta lo envíe dando un salto atrás). Debe evitarse, pues, esta posibilidad.

Además, la rutina que se propone sólo funciona con SLOW, y se detiene al pasar a FAST. Como consecuencia, el programa deberá funcionar en SLOW, y no podrá contener las instrucciones PAUSE ni LOAD, ya que durante la ejecución de éstas el aparato pasa provisionalmente a modo FAST. La grabación deberá realizarse exclusivamente en la forma que se indica más adelante.

Existen otras limitaciones. No estoy seguro de haberlas descubierto todas, de modo que el usuario deberá experimentar con el programa protegido hasta convencerse de que todo marcha bien. Mencionaré como limitación adicional, que el programa no admitirá entradas de cadenas de longitud superior a un renglón, debido a que en el momento de

```

1 REM <>6RND<>5RNDTAN LPRIN
T SGN AT LET 6RND5RND FAST E
ND PRINT VAL STR$ FAST PEEK CLE
AR EARNDCS UNPLOT <>EARNDC<> DIM
LN RND PRINT WU RNDUCODE L
ET TAN ?PILN SCROLL CLEAR AC
S VEARNDCSIN TAN EARNDC
253 PLOT LN ?RND1234567890
100 REM CLAVE DE
    DESPRIVATIZACION
105 PRINT "BUENOS DIAS..." "SOY
UN PROGRAMA PROTEGIDO..." "SI ACIE
RTA LA CLAVE SECRETA... CONSEGU
IRA DESPRIVATIZARME."
110 INPUT M$
120 IF M$="ABO" THEN GOTO 9000
125 CLS
130 PRINT "LO SIENTO..." "NO ACE
RTO LA CLAVE."
140 PRINT
150 PRINT "DE TODOS MODOS LE DA
RE OTRA OPORTUNIDAD."
160 PRINT
170 PRINT "ELIJA UNA DE LAS SIG
UIENTES OPCIONES:"
180 PRINT
190 PRINT "1. ENTRAR EN UN CICLO
O SIN SALIDA..." "2. ENTRAR EN UN
CICLO CON SALIDA"
200 INPUT X

```

```

205 CLS
210 IF X=1 THEN GOTO 300
220 IF X=2 THEN GOTO 400
230 PRINT "HA PULSADO UN NUMERO
QUE NO VALE..." "PUES AHORA ME BO
RAO EN"
240 FOR N=50 TO 0 STEP -1
245 PRINT AT 7,10;N;" "; "SEGUND
OS..."
250 NEXT N
260 NEW
300 PRINT "YA ESTOY EN UN CICLO
SIN SALIDA..." "AQUI ME QUEDO."
305 GOTO 305
400 PRINT "ESTAMOS EN UN CICLO,
PERO SE PUEDE SALIR PULSANDO
""X""
410 IF INKEY$<>"X" THEN GOTO 41
0
420 CLS
430 GOTO 170
5000 STOP
9000 REM DETENCION DEL PROGRAMA
9010 FAST
9020 SLOW
9030 STOP
9980 REM GRABACION
9985 IF USR 16575 THEN REM
9990 SAVE "PROTECTOR"
9995 IF USR 16588 THEN REM
9998 GOTO 100

```


expandirse las dos filas inferiores del espacio inferior de la pantalla, se produce una condición de error que dispara el borrado automático. Algo parecido sucede si se comete un error sintáctico al responder a un INPUT numérico.

La rutina de protección

Se muestra en el listado nº 1 como dos cadenas de códigos hexadecimales en las líneas 20 y 22, listas para cargarse en la línea nº 1, ocupada por una sentencia REM cuyo texto deberá tener al menos 90 caracteres. Se han añadido además las líneas 9980 y 9990, para facilitar la grabación.

Al ejecutar el programa cargador (RUN), el CM entra en la línea nº 1. Después se pueden borrar las líneas 20 a 70, con lo que se obtiene el listado nº 2. La línea 253 PLOT... es un «espejismo» causado por una anomalía (inofensiva) del listado, provocada por la presencia de un código de NEWLINE (76h) en medio del CM.

Hecho esto, se puede escribir el programa en BASIC. Si lo que se desea es proteger un programa ya escrito, éste debe cargarse primero, y las operaciones arriba indicadas se realizarán con el programa cargado. Naturalmente, las líneas 1 a 70 deberán estar libres, o bien el cargador se situará en otro lugar. En todo

caso, la sentencia REM del principio deberá ocupar la primera línea del programa.

Las líneas 9985 y 9995 que se muestran en el listado nº 3 se añaden para que el programa quede correctamente protegido al cargarlo. La grabación se efectuará mediante la orden: GOTO 9980.

La línea 9985 contiene una llamada a una subrutina que desarregla el archivo de pantalla. Esto impide que funcione el truco mostrado al principio para evitar la autoejecución. Si se usa dicho truco, aparecerá un CRACK.

La línea 9995 llama a otra subrutina que arregla el archivo de pantalla, y pone en marcha el mecanismo de protección.

Dicho mecanismo consiste en una comprobación periódica del estado de la variable ERRNR, que almacena el código de error en curso (decrementado en una unidad). Para ello se aprovecha el sistema de interrupciones no enmascarables (NMI), de forma similar a como se hacía en la rutina AUTO-REPEAT publicada en el nº 14 de ORDENADOR PERSONAL. Cuando el código detectado es distinto de cero, se ejecuta una llamada a la rutina de inicialización de la ROM.

El programa que se muestra en el listado nº 3 aún no está preparado para funcionar. Si se graba y se carga, se autoejecutará, pero también se detendrá con un código de error cero al alcanzar la última línea (9995). Es necesario disponer de un salto detrás de esta línea, que lleve al principio del programa.

Debe tenerse en cuenta que, si el programa ha sido diseñado apropiadamente, sólo podrá ser grabado una vez, porque después quedará protegido incluso para su autor. Por ello debe diseñarse una clave que permita, al menos al autor, desprivatizarlo. Esto es lo que se muestra en el listado nº 4. Si se introduce la cadena «ABO» en el INPUT de la línea 110, se producirá un salto a la línea 9000, y se ejecutará la secuencia FAST-SLOW que detiene la rutina de protección. La cadena clave, lógicamente, sólo será conocida por el autor del programa, y puede ser tan complicada como se quiera (siempre que no exceda de 29 caracteres).

El listado nº 5 muestra un programa protegido algo más largo. También debe grabarse mediante GOTO 9980, y su clave de desprivatización es «ABO».

direc.	CM	etiq.	mnemónicos	Comentarios	
4082	DD22BB40	RUT	LD (VI), IX	Tras una NMI el aparato ejecuta un JP (IX) que lo envía a L2.	
	DD218B40		LD IX, L2		
	C9		RET		
408B	E1	L2	POP HL	Cadena de instrucciones destinadas a que se ejecute un salto a L1 tras el refresco de la pantalla	
	D1		POP DE		
	C1		POP BC		
	F1		POP AF		
	22BD40		LD (V2), HL		
	21AB40		LD HL, L1		
	E5		PUSH HL		
	2ABD40		LD HL, (V2)		
	F5		PUSH AF		
	C5		PUSH BC		
	D5		PUSH DE	Dos instrucciones de demora (temporización fina) Salto a la rutina de visualización con HL inicializado.	
	E5		PUSH HL		
	B6		OR (HL)		
	D3FD		OUT (FD), A		
	2A0C40		LD HL, (D FILE)		
	CBFC		SET 7, H		
	DD2ABB40		LD IX, (V1)		
	DDE9		JP (IX)		
40AB	CD8240	L1	CALL RUT		Ajusta de nuevo IX. Guarda registros a usar. Ajusta contador de líneas de la pantalla.
	F5		PUSH AF		
	08		EX AF, AF'		
	3C		INC A		
	08		EX AF, AF'		
	3A0040		LD A, (ERR NR)		
	3C		INC A		
	C40000		CALL NZ, START		
	F1		POP AF		
	C9		RET		
40BB	0000	V1	DEFW 0000	Comprueba código de error. Si no es cero, llamada a inicialización. Si lo es, restaura registros, y retorno. (Aquí se almacenan datos provisionalmente). Se pone modo FAST para interrumpir visualización. Se desarregla el archivo de pantalla.	
40BD	0000	V2	DEFW 0000		
40BF	CDE 702	L3	CALL SET-FAST		
	FCDB6		RES 6, (CDFLAG)		
	2A0C40		LD HL, (D FILE)		
	36C7		LD (HL), C7		
	C9		RET		
40CC	2A0C40	L4	LD HL, (D FILE)		
	3676		LD (HL), 76		
	00		NOP		
	FDCB3BF6		SET 6, (CDFLAG)	Se pone modo SLOW.	
	CD0702		CALL SLOW/FAST		
	C38240		JP RUT		
(4082h=16514d; 408Fh=16575d; 40CCh=16588d.)					
RUTINA DE PROTECCION DESENSAMBLADA					

Miguel A. Lerma

**GARANTIA
UN AÑO**

★ **92.500**

KATSON

La mayor variedad
en tarjetas
y accesorios
para tu **APPLE***

* (APPLE) es marca registrada de Apple Computer Inc.

KATSON II

ECHATE UNOS
MARCIANITOS

JOYSTIC
por
tan sólo

5.700

ALTA
FIABILIDAD

62.500

DISK DRIVE TRACCION DIRECTA

DISK DRIVE

58.500

MECANICA SHUGART

Este verano ventila tus
tarjetas y fuentes de
alimentación por

10.000

Aprovecha las posibilidades gráficas
con el tablero gráfico PLOTT II.

17.000

Utiliza tu T.V.
con el
modulador
R.F.



3.500

estos son nuestros precios sin competencia

ORDENADORES PERSONALES

KA-001	KATSON II	92.500
KA-002	KATSON II con teclado numérico	98.500

TARJETAS Y ACCESORIOS

CD-001	8088 CARD	117.300
CD-002	A/D - D/A CARD	96.850
CD-003	A/D CARD	63.200
CD-004	IEEE-488 INTERFACE CARD	55.000
CD-005	6809 CARD	60.700
CD-006	SERIAL INTERFACE RS-232 C	14.900
CD-007	SUPER SERIAL CARD	36.750
CD-008	COMMUNICATION CARD	14.250
CD-009	128 K RAM CARD	44.000
CD-010	CP/M CARD	13.500
CD-011	WILD CARD	18.500
CD-012	GRAPPLER + BUFFER CARD	39.500
CD-013	TIME II CARD	19.125
CD-014	PARALLEL PRINTER CARD	12.375

CD-015	EPROM WRITER	16.500
CD-016	80-COLUMN CARD	16.000
CD-017	CONTROLADOR	11.300
CD-018	LANGUAGE CARD	13.500
CD-019	16 K RAM CARD	12.900
CD-020	PAL CARD	15.500
CD-021	6522 PARALLEL CARD	16.200
CD-022	MUSIC CARD	18.750
CD-023	SPEECH CARD	20.000
CD-024	80 COLUMN SOFT SWITCH CARD	34.900
CD-025	RF Modulador	3.500
CD-026	COOLING FAN	10.000
CD-027	JOYSTICK para APPLE	5.700
CD-028	SWITCHES 40/80 COLUMNAS	2.500
CD-029	TABLERO GRAFICO PLOT II	17.500

DISK DRIVER

DD-001	Disk driver - Unidad de disco flexible simple cara simple densidad 143 K. Mecánica Shugart.	58.500
--------	---	--------

**OFERTA
ESPECIAL**

97.500!!

**CONFIGURACION
"ESTUDIANTE"**

1	KATSON II con manual en castellano y cassette.	92.500
1	MODULADOR RF.	3.500
1	JOYSTICK	5.700
1	UNIDAD LECTORA DE CINTA	7.500

109.200

**CONFIGURACION
"GESTION"**

1	KATSON II c.t.n.	98.500
2	DISC DRIVER	117.000
1	CONTROLADOR	11.300
1	LANGUAGE CARD	13.500
1	MONITOR FOSFORO VERDE	29.000
1	PRINTER CARD	12.375

281.675

262.000!!

KATSON es una exclusiva de:
ANGLEX
Anglo-Española de Trading, S. A.
Ayala, 13
MADRID-1
Tels. 276 22 74
276 22 75
Telex: 42.597 ANLE

Geografía espacial

«Y sin embargo, gira» exclamaría seguramente Galileo si hubiera podido ver funcionar este programa que dibuja, sobre la impresora de un PC-1500, el globo terrestre —con sus mares y continentes— desde todos los ángulos imaginables por los cartógrafos aficionados.

¿Quién no ha notado nunca las deformaciones que aparecen en una carta geográfica debidas a la redondez de la Tierra? Los cartógrafos utilizan actualmente varios sistemas de proyección: cónica, ci-

lindrica, etc.; pero ninguno ofrece una reproducción fiable. A medida que nos acercamos a las regiones polares, las superficies geográficas se nos presentan de forma desproporcionada, y el bello pla-

nisferio se revela mucho menos preciso que el majestuoso globo terráqueo.

¿Cómo resolver el compromiso entre precisión y tamaño?

El problema parece estar resuelto con este programa para PC-1500, que permite obtener con la impresora gráfica la representación del globo terrestre desde todos los puntos de vista. Del polo al ecuador, él permite entrar las coordenadas deseadas y la impresora dibujará la correspondiente cara del globo, sin las deformaciones observables sobre las cartas geográficas clásicas.

Un único inconveniente para quien quiera usar este programa: las interminables líneas de DATA, que corresponden a las coordenadas de cada uno de los puntos de los contornos geográficos.

Meridianos y paralelos son trazados automáticamente

Por lo demás, su utilización es bien simple. El programa es interactivo y comienza con una serie de preguntas relativas al ángulo bajo el que se quiere ver la Tierra.

¿Radio? pide el radio de la Tierra en el diseño. Este comando da la talla final del dibujo. Un radio de 22 ocupa completamente el ancho del papel de la impresora, pero es posible introducir un radio mayor, con el inconveniente de que se pierde parte del diseño (ver figuras).

Alfa nos pide la entrada de un valor entre 0 y 360, que corresponde al ángulo vertical de inclinación de la Tierra sobre su eje (ver figura 1).

Beta concierne al ángulo de rotación del conjunto sobre el eje horizontal.

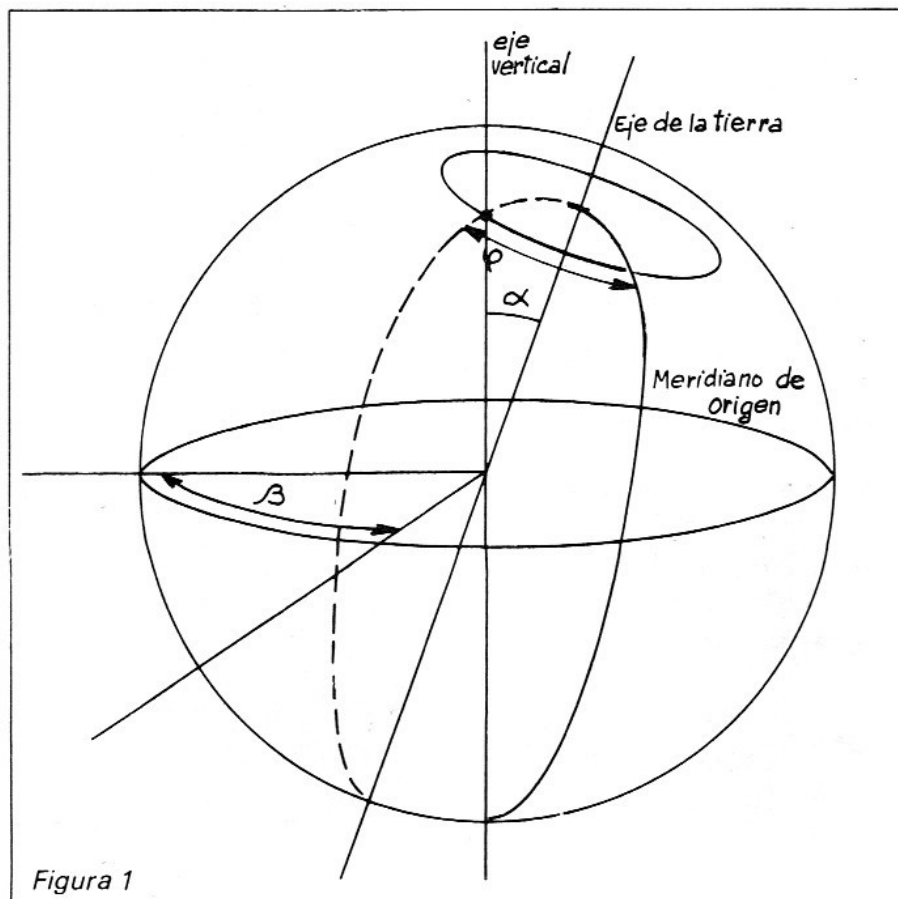


Figura 1

```

10:TEXT :WAIT 0:
DEGREE :CLEAR
20:INPUT "Radio?"
;S,"Alfa :";A,
"Beta :";B,"Ph
i :";F,"Cuadri
cula?";G,"Paso
?";P
30:INPUT "Transpa
rente (S o N)
: ";U$
40:U=1:IF LEFT$(
U$,1)="S"LET U
=2
70:INPUT "Color (
S o N) : ";CO$:
PRINT " Estoy
Dibujando!!"
80:C=0:D=0:IF
LEFT$(CO$,1)=
"S"LET C=1:D=2
90:CSIZE 2:LPRINT
"RADIO";S:
LPRINT "ALFA";
A:LPRINT "BETA
";B:LPRINT "PH
I";F:LPRINT "C
UADRICULA";G:
LPRINT "PASO";
P
100:LPRINT "HORA:"
;TIME
110:REM CONTORNO
120:COLOR D:T=9,R=
S*4.9
130:LCURSOR 0:
GRAPH :SORGN :
GLCURSOR (100,
-1.1*R):SORGN
:GLCURSOR (R,0
):T=0
140:FOR I=0TO 360
STEP P
150:X=R*COS I:Y=R*
SIN I
160:LINE -(X,Y),T
170:NEXT I
180:REM MERIDIANO
S
190:GOSUB 210
200:GOTO 230
210:D=D+C,D=D-4*
INT (D/4):
COLOR D
220:RETURN
230:T=9,J=SIN B,K=
COS B,H=SIN A,
M=COS A
240:IF G=0GOTO 470
250:FOR O=0TO 360
STEP G
260:FOR L=90TO -90
STEP -P
270:GOSUB 290
280:GOTO 370
290:Q=SIN (O-F),E=
COS (O-F),U=
COS L,W=SIN L
300:ON UGOTO 310,3
30
310:Z=R*(W*H*K-U*Q
*KJ+U*E*M*K)
320:IF Z<0LET T=9
330:X=R*(U*Q*K+W*H
*KJ+U*E*M*J)
340:Y=R*(W*M-U*E*H
)

```

```

350:LINE -(X,Y),T
360:T=0:RETURN
370:NEXT L
380:T=9:NEXT O
390:REM PARALELOS
400:GOSUB 210
410:FOR L=-90+GTO
90-GSTEP G
420:FOR O=0TO 360
STEP P
430:GOSUB 290
440:NEXT O
450:T=9:NEXT L
460:REM CONTINENT
ES
470:RESTORE 580:
GOSUB 210
480:READ N,L,O
490:ON ERROR GOTO
560
500:GOSUB 290
510:FOR I=2TO N
520:READ L,O:T=0
530:GOSUB 290
540:NEXT I
550:T=9:GOTO 480
560:GLCURSOR (-2*P
,-1.8*R):TEXT
:CSIZE 2:
LPRINT "HORA:"
;TIME :END
570:REM EUROPA
580:DATA 218,41,29
,42,35,41,38,4
2.5,42.3,46,37
,48,39,46.5,35
,46,37,44.3,34
,45.5
590:DATA 32,46.2,3
3.5,47,31,42.5
,27,41,29,40.8
,23,38,24,36.5
,22.8,40.5,19.
5,42,19.5
600:DATA 45.7,13.7
,45.5,12.3,44.
4,12.3,43.6,13
,6,42.5,14.1,4
0,18.5,40.5,17
,39.7
610:DATA 16.5,39,1
7.2,38,15.6,38
,12.5,36.6,15,
38.9,16.1,40,1
5.7,41.3,13
620:DATA 43,10.5,4
4,3,8.9,43.2,6
,2,43.5,4,42.7
,3,41.8,3,3,39
,5,-.4,38.7,.3
,36.6
630:DATA -2.1,36.5
,-4.8,36,-5.4,
37.1,-6.7,37,-
8.8,38.6,-8.8,
38.6,-9.4,41.2
,-8.6
640:DATA 43.1,-9.3
,43.7,-7.7,43.
3,-1.5
650:DATA 46.1,-1.2
,47.3,-2.5,48,
-4.7,48.6,-4.7
,48.8,-3.1
660:DATA 48.7,-1.7
,49.8,-2,49.8,
-1.3,49.4,-1.1
,49.3,-.1,49.7,
,2,50.2,1.5,5
0.9

```

```

670:DATA 1.6,51.4,
3.6,53.3,4.7,5
4.8,3,57,8.1,5
7.6,10.7,56.4,
11.9,54.5,10,5
4,14.2
680:DATA 55,20,59,
22,60,30,60.6,
28,60,22,63,21
,65.6,26,66,22
,61,17,60,19,5
6,16
690:DATA 55.4,13.5
9,10.3,58,7.6,
58.5,6,62.5,5.
5,64,10,70.3,1
9,71.2,27,67.8
,41.5
700:REM ASIA
710:DATA 66.5,39,6
7.2,33,64.5,35
,64,40,68.2,44
,69,67,72,70,7
7,112,74,110,7
2,130,70
720:DATA 175,67,19
0,66,177,63,18
0,60,170,60,16
3,59,162,51,15
7,57,156,62,16
3,62
730:DATA 157,59,15
3,59,143,55,13
5,54,141,48,14
0,39,128,35,12
9.5,34,126,39,
125.5
740:DATA 41,121,38
.5,118,30,122,
23,117,21,110,
22,108,19,105.
5,14.5,109,11.
5,109,8
750:DATA 105,13,10
0.5,9,99,5,103
.5,1,104,4,101
,9,98,17,97,23
,92,15,80,10,8
0,8,77
760:DATA 12,74.5,2
1,72,25,67,25,
56,30,50,29.5,
49,24,53,25,56
,24,56,23,60,1
7,56,12.5
770:REM AFRICA
780:DATA 44,28,35,
28,33,15,40,10
.5,45,12,51.4,
4,47.7,-5,39,-
16,41,-20,35,-
25,35,-26
790:DATA 33,-29,32
,-34,26,-35,20
,-18,12
800:DATA -11,14,-1
,9,3,10,4.6,8.
4,4.3,5.9,6.5,
4.3,4.8,-2,4.6
,-7.7,7.8,-12.
9,9.6
810:DATA -13.4,12.
4,-16.7,14.9,-
17.6,17.2,-16.
1,21.3,-17.2,2
8,-12.9,30.3,-
9.5,31
820:DATA -9.8,32,-
9.8,33.3,-8.3
830:DATA 33.9,-6.9
,35.8,-6,35.9,
-5.4,35.2,-4.7

```


, 35, -2, 36.4, 1,
37.3, 10.2
840: DATA 36.7, 10.4
, 37.1, 36.1, 10
, 5, 35.2, 11.1, 3
, 4, 10, 32.8, 12.5
, 32.94, 13.2, 32
, 4, 15.3
850: DATA 31.5, 15.6
, 30, 19, 31, 20, 3
, 2, 19.7, 33, 22, 3
1
860: DATA 29, 31.6, 3
, 1, 31.2, 33.5, 37
, 36, 37, 28, 40, 2
, 6, 41, 29
870: DATA 19, 58.5, -
, 5, 58.2, -1.8, 56
, -3.3, 56, -2, 53
, .5, 53, 1.6
880: DATA 52.2, 1.7,
, 51.3, .8, 51.3, 1
, 5, 50.9, 1, 50, -
, 5.8, 51.4, -3.7
890: DATA 51.7, -5.5
, 3.3, -4.5, 53.3,
, -3, 55, -3.5, 54.
, 7, -5, 57.5, -6.5
, 58.5, -5
900: REM ISLAS
910: DATA 5, 55.3, -6
, .5, 54.3, -10, 51
, .4, -10, 52.2, -6
, .3, 55.3, -6.5
920: DATA 7, 66.5, -2
, 2.5, 65.4, -24.5
, 66.6, -16, 65, -
, 13.5, 63, -19.64
, -22, 66.5, -22.
5
930: DATA 10, 43, 9.4
, 42.4, 8.5, 41.5
, 8.8, 40.9, 9.8,
, 39.1, 9.7, 38.9,
, 8.4, 40.8, 8.4, 4
, 1.3, 9.2
940: DATA 42.1, 9.6,
, 43, 9.4
950: DATA 6, -13, 49,
, -17, 44, -25, 44,
, -25, 47, -15, 50.
, 5, -13, 49
960: DATA 12, 60, -44
, 65, -40, 70, -22
, 82, -15, 83.6, -
, 30, 78.5, -73.76
, -68, 75.6, -59,
, 70, -51, 66
970: DATA -53.5, 61,
, -48, 60, -44
980: REM AMERICA
990: DATA 84, 63, -77
, 52, -56, 50, -65
, 46, -64, 43.7, -
, 70.4, 41.5, -70.
, 7, 40.6, -74, 37,
, -76
1000: DATA 35.2, -7
, 5.7, 31, -81.6
, 27, -80, 25, -
, 80.5, 28, -82.
, 7, 29, -82.5, 3
, 0, -84, 30.3, -
, 89, 29
1010: DATA -90, 29.
, 7, -94, 27, -97
, .5, 22, -97.7,
, 19, -96, 18.4,
, -94, 19, -91, 2
, 1, -90, 21.6, -
, 87, 16
1020: DATA -89, 15.
, 6, -83, 10.5, -
, 83.5, 9, -81.5
, 9.7, -79.8, -
, 77, 11, -75, 12
, -71, 10.6, -6
, 3, 4

1030: DATA -52, 0, -1
, 50, -36, -34, -4
, 2, -38, -22, -4
, 1, -25, -48, -2
, 8, -48, -41, -6
, 3, -51, -69, -5
, 5, -65
1040: DATA -55, -70
, -50, -76, -37
, -74, -18, -70
, -6, -81, 0, -8
, 1, 6.6, -77.5,
, 9, -79, 7, -81,
, 9.5
1050: DATA -85, 13,
, -88, 14, -91.5
, 16.2, -95, 15
, 7, -96.6, 19.
, 6, -106, 22, -1
, 05.7, 29, -112
, 4
1060: DATA 31.3, -1
, 13, 31.6, -115
, 30, -114.6, 2
, 3, -109.5, 25,
, -112.3, 30, -1
, 15.9, 34, -118
, 5
1070: DATA 34.5, -1
, 20.7, 39, -124
, 43, -124.5, 4
, 8.5, -124.5, 5
, 9, -138, 61, -1
, 48, 54, -165, 5
, 9
1080: DATA -158, 62
, -166, 68, -16
, 7, 71, -157, 68
, -110, 70, -82
, 60, -95, 54, -
, 80, 63, -77
1090: REM PACIFIC
0
1100: DATA 32, -10.
, 5, 142.4, -17.
, 5, 141, -15, 13
, 5.5, -12, 137,
, -11, 132, -15,
, 129, -14, 127,
, -20
1110: DATA 120, -22
, 114, -26, 113
, -32, 116, -34
, .5, 115, -35.2
, 118, -31.5, 1
, 30, -32.5, 133
, .5, -35
1120: DATA 135.5, -
, 33, 137.8, -35
, .2, 137.5, -38
, 140.4, -39, 1
, 43.4, -37.8, 1
, 45, -39.2, 146
1130: DATA -37.5, 1
, 50, -34.1, 151, -
, 32.7, 152.7, -
, 29, 153.6, -25
, 6, 153, -20, 1
, 48.4, -18.8, 1
, 46.3
1140: DATA -14.5, 1
, 44.7, -14.7, 1
, 44, -10.5, 142
, 4
1150: DATA 29, -63,
, -56, -64, -60,
, -66, -65, -73,
, -75, -73, -85,
, -73, -100, -75
, -100, -73, -1
, 25
1160: DATA -75, -13
, 7, -78, -165, -
, 77.6, 164, -72
, 170, -68, 155
, -66, 135, -66
, 115, -66, 90,
, -69.5

1170: DATA 75, -68,
, 70, -66, 55, -6
, 9, 40, -70, 20,
, -70, 0, -71, -1
, 0, -74, -20, -7
, 8, -35, -75, -6
, 0, -67
1180: DATA -61, -64
, 3, -59, -63, -
, 55
1190: DATA 7, 9.7, 8
, 0, 7, 82, 6.5, 8
, 1.8, 6.3, 80.5
, 6.4, 80, 8, 79
, 7, 9.7, 80
1200: DATA 22, 45.5
, 141.8, 43.3,
, 145.7, 42, 143
, 42.6, 141.6,
, 40.6, 140, 38.
, 2, 139.6, 37, 1
, 36.9
1210: DATA 35.6, 13
, 5.7, 35.6, 133
, 34, 130.9, 32
, 9, 132, 31.4,
, 131.3, 31.2, 1
, 30.2, 33.3, 12
, 9.7
1220: DATA 34, 130.
, 9, 34.5, 135, 3
, 3.5, 135.7, 36
, 140.6, 39.8,
, 142, 42.5, 139
, 7, 43.5, 141.
, 4, 45.5
1230: DATA 141.8, 1
, 1, 6, 95, 1.7, 9
, 8.8, -3.2, 101
, .6, -5.9, 105.
, 7, -6.6, 114.2
, -8.6, 114.5
1240: DATA -7.1, 10
, 5.6, -2.9, 105
, 9, .4, 103.6,
, 5, 97.5, 6, 95,
, 6, 1.9, 109.3,
, 7, 116.9, 5, 11
, 9.3, -4
1250: DATA 116, -2.
, 9, 110.3, 1.9,
, 109.3, 11, 0, 1
, 30, -2.5, 141,
, -6.5, 148, -6.
, 8, 146.8, -10.
, 7, 151
1260: DATA -7.7, 14
, 4.3, -9.3, 143
, -8, 138.4, -5
, .4, 138.1, -4,
, 133.1, 0, 130,
, 14, -34.5, 172
, 7
1270: DATA -36.7, 1
, 75.9, -37.5, 1
, 76, -38, 177.3
, -37.4, 178.5
, -41.6, 175.5
, -40.6, 172.5
1280: DATA -42.8, 1
, 71, -46, 166.2
, -46.7, 169.4
, -40.2, 175.3
, -39.3, 174, -
, 37.7, 174.8, -
, 34.5
1290: DATA 172.7

RADIO 35
ALFA 10
BETA 15
PHI 10
CUADRICULA 90
PASO 10
HORA: 111412.5535



HORA: 111413.0832

Phi pide la longitud del meridiano que pasa por el eje de la Tierra y el eje vertical; y Cuadrícula nos interroga sobre el intervalo entre los meridianos y paralelos. Si Cuadrícula=0, estas líneas no aparecen.

Paso pide el incremento con el que varía el argumento a cada paso. Un valor pequeño dará una figura más definida, pero perdemos velocidad. Por contra un valor demasiado elevado dará unas cir-

RADIO 22
ALFA 45
BETA 45
PHI 0
CUADRICULA 45
PASO 6
HORA: 121020.1525



HORA: 121020.3314

RADIO 35
ALFA 10
BETA 15
PHI 10
CUADRICULA 90
PASO 10
HORA: 111413.0944



HORA: 111413.2116

cunferencias «cuadradas». Un valor de 6 ó 7 será suficiente.

Después de estas cuestiones podemos optar en dos aspectos: Transparente, que permite visualizar la cara oculta del globo superpuesta a la cara visible, y Color, que nos dará un diseño en colores.

El principio del trazado es bien simple: meridianos y paralelos son trazados por dos bucles que calculan la latitud y la longitud automá-

RADIO 22
ALFA 10
BETA 180
PHI 0
CUADRICULA 0
PASO 6
HORA: 121017.3753



HORA: 121017.4819

RADIO 22
ALFA-49
BETA 15
PHI 10
CUADRICULA 0
PASO 10
HORA: 117.5824



HORA: 118.0841

RADIO 22.
ALFA 10
BETA 0
PHI 0
CUADRICULA 0
PASO 6
HORA: 121017.242



HORA: 121017.3446

ticamente. Los puntos del trazado están contenidos en las DATA, y una subrutina, de 290 a 360, calcula z para determinar si cada punto es visible o no (causa de la opción transparencia), luego determina x e y antes de trazar la línea.

Las fórmulas empleadas en la subrutina son simplemente los cosenos directores para el doble cambio de coordenadas debido a las rotaciones alfa (α) y beta (β).

Este programa fue optimizado para que cupiese en 6 Ko de memoria. De esta forma, con la ampliación de memoria de 4 Ko es suficiente para su funcionamiento. ☐

José Baume

Tic - Tac Tic - Tac en Vic y en Oric

VIC - 20

¿No tenéis reloj? ¡Yo tampoco! ¿Sabéis la hora que es? ¿No? Entonces, introducir este programa en un Vic 20, ponerlo en hora, y no tendréis ninguna excusa para llegar tarde a vuestras ocupaciones.

Redescubrid el viejo reloj de manecillas con este programa que utiliza los gráficos de alta resolución del Vic 20 y algunas de las funciones Basic del cartucho Super Expander.

Para trabajar con un reloj de manecillas es necesario utilizar fórmulas circulares. La función CIRCLE sólo permite el dibujo de circunferencias en trazo continuo, nosotros lo haremos con trazos discontinuos.

Dado un círculo (ver figura), las coordenadas del punto M vienen definidas por:

$$X = \sin(a) \cdot K,$$

$$Y = \cos(a) \cdot K,$$

K: radio,
A: ángulo YCM.

De esta forma, para trazar un círculo en vuestra pantalla, es suficiente elegir: el centro C (aquí 490); el radio K; el ángulo en grados que pasaremos a radianes mediante: $A \text{ (radianes)} = R \text{ (grados)} \cdot \pi(3,14)/180$.

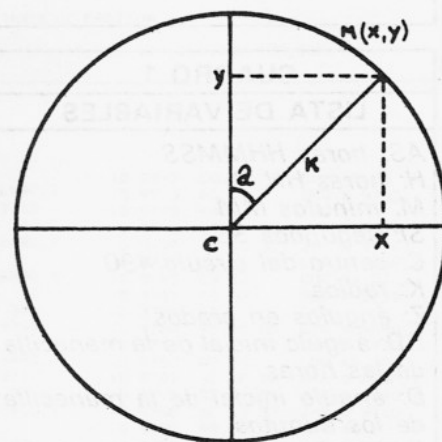
La fórmula del círculo pasa a ser:

```
FOR R=0 TO 360 STEP Z
A=R*PI/180
X=C+K*sin(A)*0,7
Y=C-K*cos(A)
POINT 2,X,Y:NEXT R
```

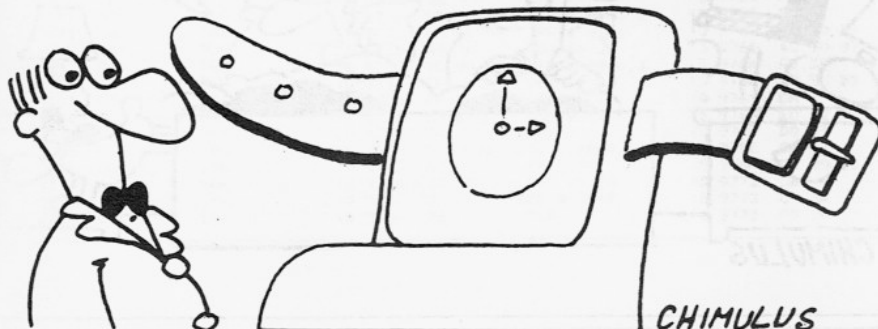
con Z igual al ángulo que nos permite el espaciado entre los puntos que constituyen el círculo.

Así por ejemplo: cuadrante de las horas $K=300$, $Z=30$ (línea 80); cuadrante de los minutos $K=330$, $Z=6$ (línea 90).

Una vez terminada la escala (líneas 8 a 90), pasemos a las manecillas. La de las horas se desplaza $Z=6$ grados cada 12 minutos. Es posicionada en las líneas 110 y 112 y comienza a moverse en la 115. De igual forma se posiciona la manecilla de los minutos en la línea 120 comenzando a girar en la 130. Y, para terminar, mediante la función TIS (líneas 140-171) se programan los segundos.



Es todo, no más problemas horarios: en todo momento vuestro ordenador os mostrará la hora. Sin embargo, los cambios horarios de invierno y verano no están programados. Pero es posible que usted mismo se anime a ello.



Programa para Vic 20

READY.

```

1 REM RELOJ DE MANECILLAS
2 REM AUTOR: RAYMOND SAXEMA
3 REM
4 REM (C) COPYRIGHT EL AUTOR Y EL O.P.
5 REM
6 REM
9 PRINT "J":POKE36878,15
10 PRINT "DAME LA HORA COMO":PRINT "HHMMSS":INPUT$
12 H=VAL(LEFT$(A$,2)):M=VAL(MID$(A$,3,2))
15 C=490:K=0:COLOR2,6,1,0
22 GRAPHIC2:CIRCLE2,0,0,C*.7,C
30 CIRCLE2,120,120,110,110:CIRCLE2,835,120,110,110:PAINT2,10,10
40 CHAR2,9,"12":CHAR17,9,"6":CHAR9,9,"9":CHAR9,15,"3":CHAR3,12,"1":CHAR5,14,"2"
50 CHAR16,12,"5":CHAR13,14,"4":CHAR15,6,"7":CHAR13,4,"8":CHAR3,6,"11"
55 CHAR5,4,"10"
60 Z=32:K=320:GOSUB200
90 Z=6:K=320:GOSUB200:R=0
110 Z=30:DD=INT(M/12)*6:IF H=12 THEN H=H-12
112 DD=Z*H+DD:TI$=A$
115 FOR I=DD TO 360 STEP 6
116 A=I*PI/180:K=170:E=C+K*SIN(A)*.7:U=C-K*COS(A)
120 D=6*M:IF M=0 THEN D=R
130 FOR P=D TO D+563 STEP 6
131 A=P*PI/180:K=250:O=C+(K*SIN(P)*.7:R=C-K*COS(P)
135 DRAW2,0,CTOE,P:CHAR17,14," "
136 DRAW2,0,CTOE,U:CHAR17,0," "
137 CHAR19,0," **--**--**--**--"
140 F=S1*6
141 A=F*PI/180:K=50:X=C+K*SIN(A)*.7:Y=C-K*COS(A)
160 POINT2,X,Y
161 S1$=RIGHT$(TI$,2):CHAR13,9,S1$:FOR L=1 TO 10:POKE36876,240:NEXT L:POKE36876,0
162 POKE36876,240:FOR L=1 TO 10:NEXT L:POKE36876,0:FOR L=1 TO 330:NEXT L
163 POINT2,X,Y:S1=VAL(RIGHT$(TI$,2))
165 IF (S1=0) OR (TI$="000000") THEN 175
171 GOTO 140
175 DRAW0,0,CTOE,P:NEXT R:M=0
180 DRAW0,0,CTOE,U:NEXT I:H=0:DD=0:M=0:GOTO 110
200 FOR P=D TO D+563 STEP 6
210 A=P*PI/180:K=240:O=C+(K*SIN(P)*.7:R=C-K*COS(P):POINT2,X,Y:NEXT:RETURN

```

READY.

CUADRO 1

LISTA DE VARIABLES

AS: horas HHMMSS
 H: horas HH
 M: minutos MM
 S1: segundos SS
 C: centro del círculo 490
 K: radios
 Z: ángulos en grados
 DD: ángulo inicial de la manecilla de las horas
 D: ángulo inicial de la manecilla de los minutos
 F: ángulo inicial del punto de los segundos
 (E,U); (O,P); (X,Y)
 Estas coordenadas sirven, a partir de C, para trazar respectivamente la manecilla de las horas, de los minutos, y el punto de los segundos.



ORIC

RELOJORIC

El programa para Oric permite obtener la representación horaria permanente a la décima de segundo. Completará, después de su adaptación, al del Vic 20.

CUADRO 2

Descripción del programa Basic

Líneas 100-300: escriben el programa ensamblador en la cabecera de la RAM.
Líneas 500 y 510: inicializan la hora.
Líneas 400 y 600: direccionan el fin de la interrupción hacia el programa reloj (reemplaze el RTI por un JUMP\$9700).
Línea 595: dobla la velocidad de repetición.

Descripción del programa ensamblador

Líneas 1000-1030: cuentan 20 interrupciones para una décima de segundo.
Líneas 1050-1060: inicializan los registros.
Líneas 1070-1110: visualizan las décimas de segundo.
Líneas 1120-1210: visualizan los minutos y segundos.
Líneas 1220 hasta el final: visualizan las horas.

Programa para Oric y...

10 REM RELOJORIC

50 AD=#230:DA=#40:POKEAD,DA

100 RESTORE

110 AD=#9700:HIMEMAD

120 AD=#9700

150 REPEAT

200 READDA

210 POKEAD,DA

290 AD=AD+1

300 UNTILDA=##FF

400 AD=#231:DA=#9700:DOKEAD,DA

500 AD=##B90

510 REPEAT:READDA:POKEAD,DA:AD=AD+1:UNTILDA=##20

560 POKE#97F0,0

595 DOKE#306,#2710/2

600 POKE#230,76

1000 DATA#48,#8A,#48,#98,#48

1010 DATA#AE,#F0,#97,#E8,#E0,#14

1020 DATA#F0,#10,#8E,#F0,#97

1030 DATA#68,#A8,#68,#A8,#68,#48

1040 DATA#00,0,0,0,0,0,0

1050 DATA#A9,0,#8D,#F0,#97,#A9,#30

1060 DATA#A2,#9

1070 DATA#BC,#90,#BB,#C8,#C0,#3A,#F0,#7,#98

1080 DATA#9D,#90,#BB,#4C,#10,#97

1090 DATA#A9,#30

1100 DATA#9D,#90,#BB,#C8,#E0,8

1110 DATA#D0,#E7,#CA

1120 DATA#BC,#90,#BB,#C8,#C0,#3A,#F0,#07,#98

1130 DATA#9D,#90,#BB,#4C,#10,#97

1140 DATA#A9,#30

1150 DATA#9D,#90,#BB,#CA

1160 DATA#BC,#90,#BB,#C8,#C0,#36,#F0,#07,#98

1180 DATA#9D,#90,#BB,#4C,#10,#97

1190 DATA#A9,#30

1200 DATA#9D,#90,#BB,#CA,#E0,2

1210 DATA#D0,#D1,#CA

1220 DATA#BC,#90,#BB,#C8,#C0,#3A,#F0,#24

1230 DATA#C0,#34,#F0,#7,#98,#9D,#90,#BB

1240 DATA#4C,#10,#97

1250 DATA#BD,#8F,#BB,#C9,#32

1260 DATA#F0,#7,#98,#9D,#90,#BB

1270 DATA#4C,#10,#97

1280 DATA#A9,#30,#9D,#8F,#BB,#9D,#90,#BB

1290 DATA#4C,#10,#97

1300 DATA#A9,#30,#9D,#90,#BB

1310 DATA#CA,#BC,#90,#BB,#C8,#98

1320 DATA#9D,#90,#BB,#4C,#10,#97

2000 DATA##FF

2050 REM CODIGO HORA DE PARTIDA

2100 DATA#32,#30,#3A,#33,#30,#3A,#30,#30,#3A,#30,#20

en ensamblador

```
:D 9700 48 PHA
:D 9701 8A TXA
:D 9702 48 PHA
:D 9703 98 TYA
:D 9704 48 PHA
:D 9705 AE F0 97 LDX $ 97F0
:D 9706 E8 INX
:D 9709 E0 14 CPX ## 14
:D 970B F0 10 BEQ $ 971D
:D 970D 8E F0 97 STX $ 97F0
:D 9710 68 PLA
:D 9711 A8 TAY
:D 9712 68 PLA
:D 9713 AA TAX
:D 9714 68 PLA
:D 9715 40 RTI
:D 9716 00 BRK
:D 9717 00 BRK
:D 9718 00 BRK
:D 9719 00 BRK
:D 971A 00 BRK
:D 971B 00 BRK
:D 971C 00 BRK
```

```
:D 971D A9 00 LDA ## 00
:D 971F 8D F0 97 STA $ 97F0
:D 9722 A9 30 LDA ## 30
:D 9724 A2 09 LDX ## 09
:D 9726 BC 90 BB LDY $ BB90, X
:D 9729 C8 INY
:D 972A C0 3A CPY ## 3A
:D 972C F0 07 BEQ $ 9735
:D 972E 98 TYA
:D 9731 F 9D 90 BB STA $ BB90, X
:D 9732 4C 10 97 JMP $ 9710
:D 9735 A9 30 LDA ## 30
:D 9737 9D 90 BB STA $ BB90, X
:D 973A CA DEX
:D 973B E0 08 CPX ## 08
:D 973D D0 E7 BNE $ 9726
:D 973F CA DEX
:D 9740 BC 90 BB LDY $ BB90, X
:D 9743 C8 INY
:D 9744 C0 3A CPY ## 3A
:D 974 F0 07 BEQ $ 974F
:D 9748 98 TYA
```

```
:D 9749 9D 90 BB STA $ BB90, X
:D 974C 4C 10 97 JMP $ 9710
:D 974F A9 30 LDA ## 30
:D 9751 9D 90 BB STA $ BB90, X
:D 9754 CA DEX
:D 9755 BC 90 BB LDY $ BB90, X
:D 9758 C8 INY
:D 9759 C0 36 CPY ## 36
:D 975B F0 07 BEQ $ 9764
:D 975D 98 TYA
:D 975E 9D 90 BB STA $ BB90, X
:D 9761 4C 10 97 JMP $ 9710
:D 9764 A9 30 LDA ## 30
:D 9766 9D 90 BB STA $ BB90, X
:D 9769 CA DEX
:D 976A E0 02 CPX ## 02
:D 976C D0 D1 BNE $ 973F
:D 976E CA DEX
:D 976F BC 90 BB LDY $ BB90, X
:D 9772 C8 INY
:D 9773 C0 3A CPY ## 3A
:D 9775 F0 24 BEQ $ 979B
```

```
:D 9777 C0 34 CPY ## 34
:D 9779 F0 07 BEQ $ 9782
:D 977B 98 TYA
:D 977C 9D 90 BB STA $ BB90, X
:D 977F 4C 10 97 JMP $ 9710
:D 9782 BD 8F BB LDA $ BB8F, X
:D 9785 C9 32 CMP ## 32
:D 9787 F0 07 BEQ $ 9790
:D 9789 98 TYA
:D 978A 9D 90 BB STA $ BB90, X
:D 978D 4C 10 97 JMP $ 9710
:D 9790 A9 30 LDA ## 30
:D 9792 9D 8F BB STA $ BB8F, X
:D 9795 9D 90 BB STA $ BB90, X
:D 9798 4C 10 97 JMP $ 9710
:D 979B A9 30 LDA ## 30
:D 979D 9D 90 BB STA $ BB90, X
:D 97A0 CA DEX
:D 97A1 BC 90 BB LDY $ BB90, X
:D 97A4 C8 INY
:D 97A5 98 TYA
:D 97A6 9D 90 BB STA $ BB90, X
:D 97A9 4C 10 97 JMP $ 9710
```

Raymond Saxema y André Thévenin

Integración por el método Simpson con el ZX Spectrum

Aunque este programa no sea un juego ni, para muchos, resulte nada divertido, reviste una gran utilidad a la hora de resolver integrales definidas en un intervalo. ¡Animo, pues, que no le va a costar ni cinco minutos introducir esta solución a sus integrales! Y, por una remota, remotísima posibilidad, no le interesa Simpson, allá usted con Chébishev.

```

10 REM Integración por Simpson
11 REM Integrales definidas.
20 REM 1983 Javier Roldán
30 REM COPYRIGHT el autor y
100 REM Obtener datos. Definir
    dx y e.
105 LET e=EXP 1
110 LET dx=1
120 INPUT "Integral?", LINE a$
130 INPUT "Primer punto del int
    ervalo?", a
140 INPUT "Segundo punto?", b
150 INPUT "Numero de divisiones
    ?", m
155 IF m<=0 OR m-INT m<>0 THEN
    GO TO 150
160 REM Hallar y(0) hasta y(m)
170 DIM y(2*m)
175 LET x=a
180 LET ix=(b-a)/(2*m)
190 FOR r=1 TO 2*m
200 LET x=x+ix
210 LET y(r)=VAL a$
220 NEXT r
230 REM La regla de Simpson se
    pone en marcha
240 LET s1=0
250 LET s2=0
260 FOR r=1 TO 2*m STEP 2
270 LET s1=s1+y(r)
280 NEXT r
290 FOR r=2 TO 2*m-1 STEP 2
300 LET s2=s2+y(r)
310 NEXT r
315 LET x=a: LET y0=VAL a$
320 LET x=b: LET y2m=VAL a$
330 LET f=(ix/3)*(y0+y2m+2*s2+4
    *s1)
340 PRINT "La solución es: "; f
350 PRINT AT 21,0; FLASH 1; PAP
    ER 2; INK 5; BRIGHT 1; "OTRA VEZ?"
360 PAUSE 0
370 IF INKEY$<>"N" AND INKEY$<>
    "n" THEN PRINT AT 21,0; "
    : RUN
380 PRINT AT 21,0; "
400 STOP
9000 SAVE "Simpson"

```

El programa avanza de una forma totalmente lineal, por lo que resulta fácil de comprender y adaptar. Sí, de adaptar, porque aunque esté concebido en BASIC Sinclair, no costará demasiado trabajo escribirlo en cualquier otro microordenador.

Lo primero que encontramos si leemos el programa (aparte de los inevitables REMs) es una definición del número e y de una variable llamada dx. La primera sólo obedece a razones prácticas, ya que —e— se presenta muy a menudo en resolución de integrales. En cuanto a la segunda, no es más que diferencial de x, dato ineluctable en una integral de-
cente.

Mediante cuadro INPUTs, el ordenador se entera de lo que queremos que haga. La función que se va a integrar se introduce mediante una variable de cadena precedida de la sentencia LINE. Al posible profano del Lenguaje del Spectrum, le diré que LINE es perfectamente suprimible; el que use este mismo ordenador comprenderá pronto por qué está ahí. En lo que se refiere al número de divisiones, esto permite aumentar o disminuir el error, según nuestras necesidades de cada momento. En realidad el número de divisiones es doble del que se introduce, con vistas a que siempre sea par. Esto reduce el error, y no creo que moleste a nadie.

En la línea 170 hay una matriz de la que hablaremos más tarde. En la 175 inicializamos el valor de x, haciéndole corresponder el del primer punto del intervalo. Más tarde definiremos ix (incremento de x) y hallamos, gracias a un bucle, los diferentes valores de y. En la línea 230 será mejor que hagamos caso al REM y escribamos la fórmula de las parábolas:

$$\int_a^b f(x) dx \approx \frac{b-a}{6m} [y_0 + y_{2m} + 2(y_2 + y_4 + \dots + y_{2m-2}) + 4(y_1 + y_3 + \dots + y_{2m-1})]$$

Como se deduce claramente, s1 es la sumatoria de los valores y con subíndice impar y s2 la de los de subíndice par. Ya con esto, solo falta hallar y0 e y2m usando la función VAL, que ya nos sirvió antes. Y el programa queda totalmente desvelado.

Para finalizar, las líneas 350 a 380 son meros adornos.

Javier Roldán (Granada)

Nota de la redacción:

Hemos añadido en este programa las líneas 400 y 9000 para facilitar su grabación, y la 155, que asegura que el número de divisiones sea un número entero y positivo.

Quilopus

o cómo pasarlo bien matando gusanos

¿Quiere probar su habilidad matando gusanos? Gracias a su Atom, podrá pasar buenos ratos divertidos con este programa. Además sin alterar con ello el equilibrio ecológico.

Pero, cuidado puede ser invadido con gran facilidad y sólo dispone de 3 vidas.

En este juego debe usted destruir una plaga de gusanos que han invadido una plantación de setas; cuenta para ello con tres fumigadores en cada partida.

Las rutinas críticas del programa han sido realizadas en ensamblador para que el juego tenga el mayor realismo posible. La descripción del juego es sencilla: El gusano aparece en la parte superior de la pantalla, que está llena de setas. Los disparos pueden alcanzar a una seta, (accionados por la tecla J), en cuyo caso destruyen media, o al gusano. En este último caso el segmento alcanzado se convierte en seta y las otras dos mitades cobran vida independiente. Los segmentos, al encontrar una setita en su camino, bajan y cambian su sentido de marcha.

Para mover el cañón «fumigabichos» se usan las teclas CTRL y SHIFT, para lograr independencia respecto a la tecla de disparo. Si los gusanitos han logrado sobrepasar su barrera de disparos, aún queda una esperanza. Pueden ser destruidos por el viejo método de pisotearlos, (al jugar con el ATOM se pisotean con el cañón, claro). Esta facultad desaparece cuando ha matado demasiados y el cañón se torna oscuro.

Un fumigador es destruido cuando algún segmento llega a salirse de la pantalla. Al perder los tres con los que cuenta, un mensaje le indica los puntos conseguidos y cómo seguir jugando. Si alguien consigue lo que yo no he conseguido nunca (vencer todas las oleadas), entonces el programa se rendirá al faltarle recursos para ganar a semejante fumigador.

El programa cuenta con sonidos de crecimiento de seta, (un «clac»), de muerte de gusano, y de victoria sobre una oleada.

Si se ha vencido una oleada, en la siguiente aumenta el número de segmentos. Recuerde que si el cañón se vuelve oscuro pierde la facultad de pisotear.

El diseño del cañón está en el vector S, el del segmento de gusano en el K, y el de la seta en el Q.

Echemos un vistazo a las rutinas en ensamblador: TT6 realiza el movimiento de todos los segmentos, uno a uno. Se vale de HHO, que primero examina si el segmento se va a dar con la pared o con una seta.

En ese caso se vale de DDO para bajarle y cambiarle el sentido de marcha. HHO también examina si el bicho tiene una bala encima, en cuyo caso llama a TTO, que

pone una seta encima y actualiza la lista de segmentos.

La lista de los segmentos está formada por el vector M para los bytes bajos, N para los altos y F para los sentidos de desplazamiento.

WWO es una rutina usada en varios puntos para poner blanco bajo la dirección señalada en # 86 y # 87.

DD4 lee las teclas CTRL y SHIFT y mueve el cañón, previa comprobación de que no se sale por las paredes, en el sentido indicado.

NNO es la rutina que mueve las balas. Primero comprueba que hay balas que mover; luego tiene cuidado de ver si la bala va a dar a una seta, en cuyo caso actúa NN2, o a un segmento, donde actúa NN7 que deja la bala en donde está (ya se enterará el bicho de que está herido).

En ambos casos se actualiza la lista de direcciones de las balas mediante NN4. Las direcciones de las balas están en página cero a partir de # 90.

DD10 comprueba si se ha disparado, y, en ese caso, añade una bala más a la lista de balas.

La rutina ZZ0 se encarga de los sonidos, y se le pasa la frecuencia en X y la dirección en Y.

Si se quiere modificar el número de setas en pantalla, hay que modificar el bucle de la línea 2100.

Para modificar el número de disparos que se puede hacer hay que ir al bucle principal del programa (líneas 2340 y 2520) y modificar la 2400. Por cada dis-


```

>>L.
10REM
11REM          - GUSANOS -
12REM  JESUS AYUSO LENO
13REM  (C) EL AUTOR Y EL ORDENADOR PERSONAL
14REM
15REM
20DIMHH3,WW2,DD13,TT10,NN7,K7,S6
40DIMZZ8
60DIMF30,N30,M30,R0,C1,L0,Q9
80DIMP-1
100P.$21
120F.I=0 TO 90;HHI=P;N.
140F.I=0 TO 1;DIMP-1;I
160:HHO LDY@64;LDX@4
180:HH1LDA(#86),Y;LDY@0;CMP@255
200BEQ DD0
220:HH2LDA(#86),Y;CMP@20
240BNEHH3;JMPTT0
260:HH3LDA(#86),Y;CMP@20;BEQTT0
280LDAK,X;STA(#86),Y;TYA
300CLC;ADC@32;TAY
320DEX;BNE HH2
340JMPTT8
360:DD0LDA#70;CMP@1;BNEDD1
380LDA@255;STA#70
400JMP DD2
420:DD1:LDA@1;STA#70
440:DD2
460CLC;LDA#86;ADC@128;STA#86;BCCDD3;INC#87
480:DD3CLC;LDA#70;ADC#86;STA#86
500JMPHH2
520:WWOLDY@0
540:WW1
560LDA@0;STA(#86),Y
580TYA;CLC
600ADC@32;TAY
620CPY@32*4
640BNEWWI;RTS
660:TT6LDA#71;STA#71
680:TT7LDX#71;LDAM,X;STA#86
700LDAN,X;STA#87;LDY@0;LDA(#86),Y;CMP@20;BEQTT10
720JSRWWO;LDX#71;CLC;LDAM,X
740ADCF,X;STA#86;STAM,X
760LDA F,X;STA#70
780JMPHH0;TT8 LDX#71;LDA#70;STAF,X
800LDA#86;STAM,X;LDA#87;STAN,X
820DEC#71;TT9LDA#71;BNE TT7
840RTS;TT10JMPTT0
860:DD4LDA#88;STA#86
880LDA#89;STA#87
900LDA#8001;CMP@191;BEQDD5
920CMP@127;BEQDD6;RTS
940:DD5LDY@31;LDA(#88),Y
960CMP@255;BEQDD7;DEC#88;JSRWWO;JMPDD8
980:DD6LDY@1;LDA(#88),Y;CMP@255
1000BEQDD7;INC#88;JSRWWO
1020:DD8LDX@4;LDY@0;DD9LDAS,X
1040STA(#88),Y;TYA;CLC;ADC@32;TAY
1060DEX;BNEDD9
1080:DD7RTS
1100:NN0LDXL;BEQDD7;NN1LDA(#90,X)
1120LDA@0;STA(#90,X);SEC
1140LDA#90,X;SBC@128;STA#90,X
1160BCSNN3;DEC#91,X
1180:NN3LDA#91,X;CMP@79
1200BEQNN4;LDA(#90,X);CMP@255
1220BEQNN2
1240CMP@65;BEQNN7;CMP@40;BEQNN7
1260CMP@60;BEQNN2
1280LDA@20;STA(#90,X)
1300:NN5 DEX;DEX;BNENN1;RTS
1320:NN2LDA#90,X
1340STA#86;LDA#91,X;STA#87;JSRWWO
1360:NN4STX#2800;NN6LDA#92,X;STA#90,X
1380LDA#93,X;STA#91,X;INX;INX
1400CPX L;BCCNN6;LDX#2800
1420DECL;DEC L
1440JMPNN5;NN7:LDA@20;STA(#90,X);JMPNN4
1460:DD10JSR#FE71;CPY@3
1480BNEDD11;LDX L;INX;INX
1500LDA#88;STA#90,X
1520LDA@88;STA#91,X;STX L
1540LDX@40;LDY@20
1560JSRZZ0
1580:DD11RTS
1600:TT0LDY@0;LDX@0
1620:TT1LDA@,X;STA(#86),Y;TYA;CLC
1640ADC@32;TAY;INX;CPX@5
1660BNETT1;INCC;BNETT2
1680INCC+1
1700:TT2LDX#71;TT3LDAM+1,X;STAM,X;LDAF+1,X;STAF,X
1720LDAN+1,X;STAN,X;INX
1740CPXR;BCCTT3;DEC;INC#71
1760LDA@30;STA#2801;ZZ5LDY@20
1780INC#2801
1800LDX#2801;CPX@70;BEQZZ6
1820JSRZZ0;JMPZZ5
1840:ZZ6RTS
1860:ZZ0STX#2800;LDA@0
1880:ZZ1LDX#2800
1900:ZZ2DEX;BNEZZ2;DEC#2800
1920EOR@4;STA#B002
1940DEY;BNEZZ1;RTS
1960;N.;P.$6
1980V=3;H=#821F
2000?Q=60;Q?1=255;Q?2=255;Q?3=60;Q?4=60;Q?5=60
2020!#86=#8000
2030?K=65;K?1=65;K?2=40;K?3=40;K?4=65;K?5=65
2040?C=0;?C+1=0
2050?S=170;S?1=170;S?2=170;S?3=20;S?4=20;S?5=20
2060?L=0;?R=13;E=13
2080wCLEAR3;COLQURO
2100F.I=0T040;X=80;Y=6;LI.ZZ0
2120A=#8100+(A.R.%19)*128+A.R.%29
2140F.J=0T05*32 S.32;A?J=Q?(J/32);N.;N.
2170F.I=#8620 TO H STEP-#200;#88=I;LI.DD5;N.
2180!#88=#8B10
2200F.J=0T0 30
2210M?J=J;N?J=#80;N.
2220F.I=0 TO29;F?I=1;N.
2240?#8020=255;?#802F=255
2260F.I=#8000 TO #8C00 S.32;?I=255;I?30=255;N.
2300LI.DD4
2320Q=0;LI.DD5
2340DD0
2360WAIT
2380LI.TT6;LI.DD4;LI.NNO
2400Q=0+1;IFQ%3=1 LI.DD10
2420REM
2440WAIT
2460LI.NNO;LI.NNO
2480WAIT
2500LI.DD4;LI.NNO
2520U.N?1>#8B OR?R=0
2540IF?R=0 G.r
2550?R=E
2600V=V-1;IF V<=0 GOTO y
2605H=H+200
2610GOTO w
2615yP.$12"ha sido invadido""para jugar otra,"
2616P."pulse una tecla""puntos ",?C+?(C+1)*255
2620LINK #FFE3;?C=0;?(C+1)=0
2640G.1980
2660r?R=E+3;IF?R>296.1
2680E=?R;IF ?C>70 S?3=60;S?4=60
2720F.X=30T070;Y=99;LI.ZZ0;N.
2740G.w
27601P.$12"HA GANADO""NO QUEDAN MAS"
2820E.

```

paro acertado se acumula un punto en ?C, y en C?1.

Aunque no están previstas puntuaciones superiores a 256, se hace así para modificar lo que se quiera.

Respecto al juego hay que

advertir que unos segmentos se pueden montar en otros y al disparar encontramos con que matamos a uno y salen uno o dos más. Si se quiere dar sensación de que el gusano mueve las patitas se debe añadir la línea:

2470 K?1 = 65; K?4 = 65; IF 0%2 K?1 = 4; K?4 = 4

Después de una horita de teclear ya se puede empezar a matar gusanos y a partir setas.

Jesús Ayuso Leno

TRADUCTOR MONITOR/INTERPRETE para MZ-80 B

Para programar tenemos que utilizar constantemente el inglés. No es cómodo para muchos. Os propongo este mes un programa que os permita cambiar con facilidad las instrucciones BASIC por palabras del idioma que os guste más.

CAMBIO DE LAS INSTRUCCIONES BASIC, COMANDOS MONITOR Y MENSAJES SISTEMA

Debido a que tanto el Monitor como el interpretador BASIC del Sharp MZ 80B se cargan en RAM es posible modificar también la sintaxis de las instrucciones BASIC, los comandos del monitor y los mensajes del sistema. Por ej.: Cambiar la instrucción INPUT por ENTRE, ó bien el mensaje de "SET TAPE" por "CINTA ?" o incluso los carteles del directorio del disco (BTX, BSD, VOLUME No.,...).

Estas modificaciones se podrían realizar "manualmente" desde el monitor (Cambiano el contenido de las posiciones de memoria corresp.) ó desde BASIC a base de POKes. Pero para que resulte más cómodo se puede utilizar un programa como el que os adjunto.

En el programa se solicita el tipo de modificación = "MENSAJE / INSTRUCCION?" ya que, si bien se almacenan en la misma forma, el último carácter de cada instrucción está almacenado en el interpretador con el código en REVERSE del símbolo correspondiente. De esta forma por

programa se puede cambiar, cuando se trata de una instrucción, el último carácter por el correspondiente en reverse. (ASC=ASC+128).

Otro dato que se debe de introducir es la dirección de inicio de búsqueda. Conociendo la zona que debemos "explorar" ahorramos mucho tiempo si introducimos como dirección de inicio la correspondiente a esa zona. Por ejemplo = Las instrucciones de BASIC (La tabla de instrucciones del interpretador) tienen su inicio en la dirección decimal 5700, si introducimos esta dirección de inicio para la búsqueda de una instrucción el tiempo que tardará el ordenador en darnos la respuesta será muy inferior al que invertiría si la búsqueda se iniciara en la posición 0000.

Una consideración importante: Esta controlado por programa que la longitud de la variable que va a reemplazar a la instrucción o cartel sea idéntica a la original; si introdujeramos una de mayor longitud destruiríamos los datos de las memorias que hubiera a continuación, con resultados imprevisibles.

Una vez entrados los datos de la instrucción a localizar y la direcc. inicial aparece en la parte inferior de la pantalla un contador de las posiciones que se están procesando. Localizado el string se visualiza la posición de memoria correspondiente a la localización del primer carácter y aparece el mensaje "REEMPLAZAR POR? =" indicando que debemos introducir la nueva instrucción o mensaje. (ver ejemplo de ejecución).

Hay que tener en cuenta que determinadas instrucciones precisan del paréntesis o del símbolo "\$" (Ej.: SIN (, LEN (RIGHT\$) y por ello al solicitar la búsqueda se debe de entrar también estos símbolos.

Para ahorrar tiempo de ejecución he resuelto el programa de forma que localiza primero solamente el primer carácter de la variable (Rutina de LOCALIZAR RAPIDO 1^{er} CARACTER) y una vez localizado este comprueba que los siguientes correspondan al resto de la variable. De no ser así continúa a partir de la dirección siguiente a la de localización del 1^o en búsqueda nuevamente del 1^{er} carácter.

Como ya es habitual en mis programas podéis encontrar la variable Z3\$, variable utilizada por la instrucción MUSIC para emitir una serie de tres pitidos cortos cuando se produce una entrada incorrecta.

COMO CONSEGUIR GUARDAR UN JUEGO DE INSTRUCCIONES BASIC MODIFICADOS

Como es lógico una vez se desconecte el microordenador o se realice un

```
*** _TRADUCTOR MONITOR/INTERPRETE ***
```

```
INSTRUCCION ? = INPUT
```

```
LOCALIZADO EN POSICION (Dec.) = 5883
```

```
REEMPLAZAR POR ? = ENTRE
```

```
████████████████████ 5883 ██████████
```



```

1 REM -----
2 REM ---- T R A D U C T O R ----
3 REM ---- MONITOR / INTERPRETER ----
4 REM -----
5 REM ---- Autor = J.M. VIDAL LACASA ----
6 REM ---- Copyright EL AUTOR Y ----
7 REM ---- EL ORDENADOR PERSONAL ----
8 REM -----
9 REM
10 Z3$="B1R0+B1R0+B1R0":TEMPO 7
15 CONSOLE 50,24,C40
20 PRINT "*** TRADUCTOR MONITOR/INTERPRETE ***"
80 CURSOR 0,10:INPUT "MENSAJE O INSTRUCCION ? (M/I) = ":T$
90 IF (T$="M")+(T$="I") THEN 100
95 MUSIC Z3$:GOTO 80
100 CURSOR 0,10:PRINT STRING$( " ",39)
110 C$="MENSAJE ":IF T$="I" THEN C$="INSTRUCCION "
120 CURSOR 0,10:PRINT C$:INPUT "? = ":A$
125 CURSOR 0,12:INPUT "POSICION INICIO BUSQUEDA = ":IN
130 IF (IN<0)+(IN>65400) THEN MUSIC Z3$:GOTO 125
135 A$=LEN(A$):IF T$="M" THEN 160
140 U$=RIGHT$(A$,1):U$=CHR$(ASC(U$)+128)
150 A$=LEFT$(A$, (A$-1)) + U$
160 REM ---- LOCALIZAR RAPIDO 1er CARACTER ----
170 X=ASC(LEFT$(A$,1)):N=IN
180 IF PEEK(N)=X THEN 250
185 N=N+1:CURSOR 10,20:PRINT "BUSCANDO":N$=""
190 GOTO 180
250 REM ---- LOCALIZADO 1er CARACTER ----
255 Q=0:L=N:S=1:FOR P=L TO (L+A$)
260 IF CHR$(PEEK(P))=MID$(A$,S,1) THEN Q=Q+1
265 S=S+1:NEXT P:IF Q=A$ THEN 300
270 N=L+1:GOTO 185
300 REM ---- LOCALIZADO ----
305 CURSOR 0,12:PRINT "LOCALIZADO EN POSICION (Dec.) = ":L
310 CURSOR 0,14:INPUT "REEMPLAZAR POR ? = ":N$:NW=LEN(N$)
315 IF NW>A THEN MUSIC Z3$:GOTO 310
320 IF T$="M" THEN 350
325 U$=RIGHT$(N$,1):U$=CHR$(ASC(U$)+128)
330 N$=LEFT$(N$, (NW-1)) + U$
350 S=1:FOR P=L TO (L+A$-1)
355 X=ASC(MID$(N$,S,1)):POKE P,X:S=S+1
360 NEXT P:GOTO 10

```

BOOT (Carga del sistema) todas las modificaciones entradas desaparecerán. Si os interesa crear un nuevo juego de instrucciones, por ejemplo en castellano (dentro de las limitaciones de la longitud original) y poder acceder a él sin tener que modificar de nuevo mediante el programa, seguir los siguientes pasos:

- 1.º Modificar las instrucciones con el programa TRADUCTOR.
- 2.º Pasar a MONITOR con la instrucción MON.
- 3.º Colocar una cinta de cassette virgen y rebobinadla.
- 4.º Ahora grabaremos una parte del interpretador en la cinta, en OBJ. para ello una vez en monitor realizaremos las siguientes operaciones =

(Nota = estando en Monitor debe aparecer un asterisco como Prompt del sistema)

- introducimos una "S" (letra ese) y pulsamos ENTER (Comando SAVE)

- aparecerá en pantalla "FILE NAME: " indicando que debemos introducir el nombre del programa (16 car.) como ejemplo introduciremos INTERPRET y pulsamos ENTER.
- aparece ahora "S-ADR.\$" (START ADDRESS, dirección de inicio) e introducimos 1688 (Dirección inicio tablas instrucciones).
- a continuación responderemos a los mensajes de dirección de fin y

```

MON
*S
FILE NAME: INTERPRET
S-ADR.$1688
E-ADR.$18A8
J-ADR.$12B0
SET TAPE
*
*J
J-ADR.$12B0
Ready
COPY/P1

```

dirección de salto (o ejecución):
E-ADR.\$ (END ADDRESS) introducimos 18A8
J-ADR.\$ (JUMP ADDRESS) introducimos 12B0 (dirección de salida del BASIC para que, una vez cargado, el programa salte al READY de BASIC)

- Si no tenemos colocada la cassette aparece el mensaje SET TAPE, caso contrario se graba la zona que le hemos especificado en el cassette, con el nombre de programa que le habíamos indicado.
- Una vez grabado podemos verificar que la grabación ha sido correcta con el comando de monitor: "V" (VERIFY)
*V
FILE NAME: INTERPRET
FOUND INTERPRET
VERIFYING INTERPRET
OK (Si todo es correcto aparece el mensaje de OK)

Una vez grabado en cassette el programa en OBJ si se dispone de unidad de diskettes se puede grabar el programa en disco utilizando la utilidad del sistema). Colocamos el disco MASTER (o SUBMASTER) en FD1, e introducimos: RUN "Filing CMT" (o bien un RUN directamente del directorio), colocamos el cassette, al principio de cinta, un diskette en la unidad de discos FD2 (donde se grabará el programa).

Al ejecutar el programa Filing CMT aparecerá en pantalla una serie de mensajes y nos solicitará la introducción de la unidad de discos en la que se va a operar, como el diskette lo tenemos puesto en FD2 contestaremos con un 2 a la pregunta. Se pone en funcionamiento el cassette, se carga el programa y una vez cargado se graba en el disco. Finalizada la operación pulsaremos B para realizar un nuevo BOOT del sistema.

Como leer el programa que hemos grabado: Si lo tenemos en disco, podemos modificar el programa AUTO RUN para que se cargue automáticamente al realizar la carga del sistema. Para ello basta cambiar la línea de NEW por: RUN "INTERPRET" (o el nombre con que lo hubiéramos grabado)

-Si por el contrario no se dispone de diskettes y el programa está almacenado en cinta el programa se debe cargar desde MONITOR cada vez que se conecte la máquina. Una vez cargado el BASIC pasamos a MONITOR (MON), colocamos el cassette y utilizamos el comando L (LOAD) cuando nos solicite el nombre del programa podremos omitirlo, pulsando ENTER, si el programa está al comienzo de la cinta.

José María Vidal Lacasa

GESTION FAMILIAR

Cuando el ordenador administra su tesorería.

Todo empezó con la lectura de los anuncios de pisos. Los que me gustaban estaban fuera del alcance de mis posibilidades económicas; y los que sí pensaba poder comprar, eran demasiado pequeños.

La única solución era comprar un piso de tipo medio, pero en mal estado, y hacer obras, para arreglarlo poco a poco.

Pero esto significaba tener que plantearme periódicamente la misma pregunta: ¿Cuánto puedo pedir prestado e ir devolviendo cada mes, sin tener que alimentar a mi familia sólo con pan y cebolla?.

Por eso empecé el estudio detallado de los gastos e ingresos familiares. En cuanto a estos últimos, siendo asalariado, no hay demasiados problemas ni imprevistos. Un sueldo, al que se añaden las pagas extras y algunas "primas" más o menos periódicas.

Antes de planificar gastos "extras", es imprescindible conocer bien los gastos "fijos" que son muy distintos, tanto en importe como en periodicidad.

Un estudio del pasado reciente permite evaluar y fechar los distintos elementos.

• **Vivienda:** un alquiler (o la amortización de un préstamo) más los gastos de comunidad y los impuestos asociados (son 4 o 5 para una vivienda).

• **Alimentación:** las compras realizadas en tiendas son bastante difíciles de calcular con exactitud, sobre todo cuando se paga en efectivo. Pero, vista la importancia de este capítulo, hay que examinarlo muy cuidadosamente (sin olvidar los precios de las comidas en el comedor de la empresa y en colegios).

• **Servicios:** teléfono, luz, gas y quizá agua. Al menos una factura al mes, más la asistenta.

PREVISION ECONOMICA ANUAL
DESEA CREAR O MODIFICAR (C,M) ? M
INGRESOS 1 A 6 GASTOS 7 A 12

DESEA MODIFICAR CONCEPTOS ? N

INGRESOS 1 A 6 GASTOS 7 A 12

DESEA MODIFICAR CONCEPTOS ? N

CALCULANDO

B A L A N C E

#	CONCEPTO	1	2	3	4	5	6
=====	=====	=====	=====	=====	=====	=====	=====
1	INGRESOS	1	2	3	4	5	6
2	REVISTA	1	1	1	1	1	1
3	EXTRAS	2	1	2	1	2	1
7	GASTOS	2	1	1	3	3	3
8	HIPOTECA	0	0	4	0	0	4
13	SALDO ANT	0	2	5	6	9	14
14	INGRESOS	4	4	6	6	8	8
15	GASTOS	2	1	5	3	3	7
-----	-----	-----	-----	-----	-----	-----	-----
16	SALDO MES	2	5	6	9	14	15

• **Vacaciones:** viajes, alojamientos, comidas para los miembros de la familia (padres e hijos, no siempre siguen el mismo itinerario).

• **Vestir:** para cada uno de los miembros de la familia. Las compras importantes, pagadas con tarjeta de crédito, son fáciles de analizar. Las compras normales, pagadas en efectivo y las compras hechas en grandes almacenes, pueden verse englobadas en el capítulo alimentación.

• **Coche** (o coches): compra (o pago de mensualidades), garage, mantenimiento (gasolina, engrase, neumáticos. . .).

• **Niños:** importe de los colegios.

• **Ocio:** libros, revistas, periódicos, música, bricolage, regalos (Navidad y cumpleaños).

• Y no olvidar **los impuestos:** fáciles de localizar pero difíciles de prever, sobre todo más allá de 12 meses (los pagos aumentan cada año o principios del mismo).

Al hacer la suma de todos los gastos, aparecen algunos "picos" en determinadas fechas. Esto confirma que la gestión con horizonte mensual ocasiona, inexorablemente, problemas de tesorería.

Sin embargo, hay que arreglárselas para disponer siempre de un saldo positivo. Es más, si se es muy prudente, conviene que ese saldo sea superior a una determinada cantidad, para poder afrontar problemas imprevistos: accidente o enfermedad grave. . .

Previsiones difíciles.

Así, con un minucioso examen de las operaciones registradas en las cuentas corrientes o cartilla de ahorro, se pueden conocer las fechas y los importes de los distintos conceptos, tanto de los ingresos como de los gastos. Es posible, pues, sin demasiada dificultad, hacer la lista y el registro de vencimientos de todos los gastos imprescindibles, así como de los recursos disponibles. Con o sin calculadora de bolsillo (su presencia disminuye el número de errores) se puede hacer una previsión bastante correcta —tal día debe ingresarse tal cantidad, tal día tanto. . .— y, conocer la cantidad disponible en función del tiempo.

Al obrar así, vemos que, junto a los gastos claramente definidos, se hacen multitud de pequeños gastos a lo largo del año, y éstos no aparecen en ninguna cuenta por su poco importe: billetes de metro, desayunos, correo, periódicos, tintorería, cine, taxis, cuestaciones, propinas. . .

Son cantidades que corresponden a gastos en efectivo, de los cuales hay que excluir los gastos de alimentación, ya que si no se llegaría a una cantidad importante, difícil de controlar.

Para todos estos gastos, una buena solución puede ser atribuirles una cantidad global diaria.

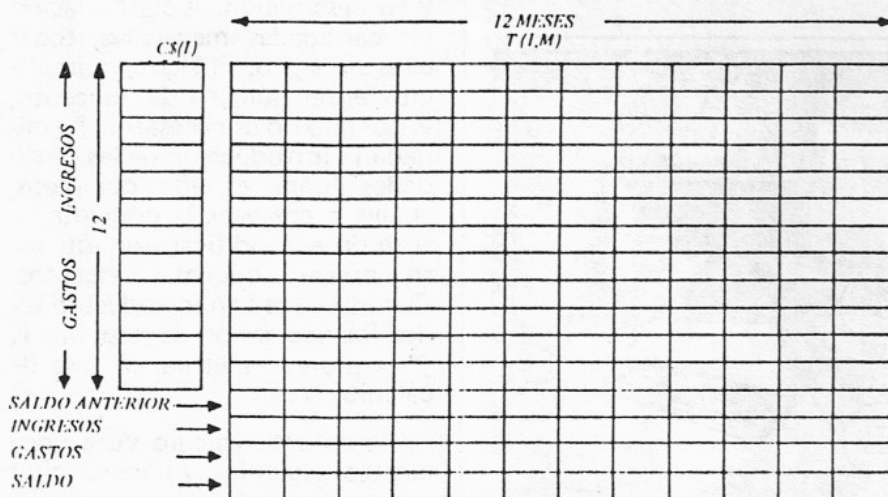


Tabla C\$ de conceptos y T. de cantidades por concepto/mes
Figura 1



El ordenador, chivo expiatorio

Esto se logra añadiendo a todos los conceptos anteriores, un gasto "constante", que permite ajustarse a la realidad, pero que no simplifica los cálculos (¿Cuántos días hay entre el 23 de marzo y el 7 de abril?).

Operando así, se obtienen resultados bastante buenos, tanto mejores cuanto más precisas sean las hipótesis.

Conociendo los ingresos y gastos "normales", se puede iniciar el estudio de los gastos "extraordinarios" —o de una variación de los ingresos—. En el ejemplo citado al principio, se trataba de trabajos de reforma del piso.

Entonces es cuando los cálculos se vuelven fastidiosos, porque es necesario hacer varias hipótesis, antes de encontrar una solución aceptable. Además, las "circunstancias" evolucionan: este año, se renuncia al arreglo de la cocina, previsto para el mes de junio (porque la nevera y la lavadora están agonizando), y se decide aprovechar las rebajas coincidiendo con el "Salón de la Electrificación" (en marzo).

¿Se pagará al contado, o en 6 ó 12 letras? Según la hipótesis, hay que modificar todos los cálculos.

Nos damos cuenta de lo pesado que es administrar, de la mejor forma posible, un presupuesto que está siempre en el límite del equilibrio.

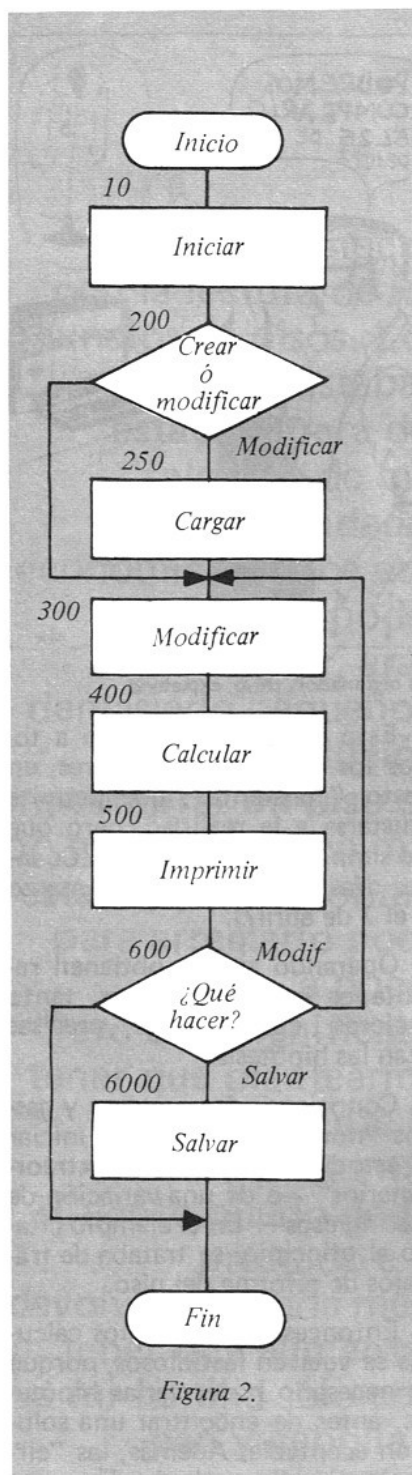


Figura 2.

Fué al percatarme de la dificultad de esta segunda fase, cuando decidí gestionar la previsión presupuestaria en un ordenador personal, adaptando el método que, a mano, me dió buenos resultados. La máquina se encarga de la parte engorrosa del trabajo, las máquinas sirven para eso.

El programa está basado en unas tablas, representadas en la figura 1. De estas tablas, puede direccionarse cada casilla, para realizar correcciones y pueden rescatarse en diskette. En una

ejecución posterior del programa, puede cargarse la tabla desde el diskette, para modificar después valores y proceder a una nueva planificación de gastos. Si resulta aceptable, vuelve a rescatarse, para que sea base de partida para la simulación siguiente. Si no es aceptable la simulación, o bien se recurre al diskette para volver a situación inicial, o bien se ajustan las previsiones. Con una ligerísima modificación, es aceptable la utilización de cassette en el caso de que se disponga de instrucción de salvado y lectura desde esta, de tablas (Tipo STORE y RESTORE). Esta idea, aunque muy limitada en este programa, es la que inspira el conocidísimo VISICALC. Es un programa que permite gestionar tablas y cuadros con facilidad, una vez definidas las operaciones a realizar entre columnas y entre filas. Hecha la definición, pueden modificarse valores, que quedarán repercutidos en los lugares adecuados, al indicar que se recalcula la tabla.

Debo de explicar con detalle las tablas CS y T y su utilización. Hay diferentes maneras de resolver el problema. Normalmente se hubiesen utilizado más de dos tablas. He querido utilizar las menos posibles, por interés didáctico. Siempre pueden no hacerme caso y utilizar 6 ó 7 tablas. ¿Y por qué 2 y no 1? Pues porque la descripción de los conceptos, tratándose de un literal, se lleva mal con las cantidades. Sólo por eso.

CS Tabla de conceptos. Tiene 12 elementos, los 6 primeros, para conceptos positivos, INGRESOS y los 6 últimos para

conceptos negativos, GASTOS. Se han fijado estos números, como podían haber sido otros. Si necesitan una dimensión mayor, sólo tienen que modificar la DIM y los bucles FOR.

T Es una tabla de dos dimensiones que contiene diferente información, según el valor del primer índice. El segundo varía entre 1 y 12, coincidiendo con los meses del año.

El primero varía entre 1 y 16 y su sentido es el siguiente:

- **T (1,M) a T (12,M)** contiene el valor del ingreso o del gasto para el mes M.
- **T (14,M)** es la suma de ingresos para el mes M. Esta totalización se consigue en 430.
- **T (15,M)** es la suma de gastos, para el mes M. Esta totalización se consigue en 440.
- **T (16,M)** es el saldo para el mes M. Ver 4200.
- **T (13,M)** es el saldo del mes anterior al M, es decir del M-1.

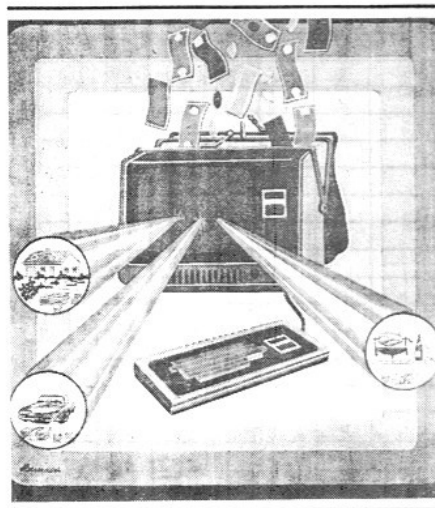
Operativa.

Disponemos, ya se ha visto, de una tabla donde hacer 6 anotaciones por mes, de ingresos y 6 de gasto.

Supongamos que es la primera vez que se ejecuta el programa. Por lo tanto, los datos anteriores no existen. Contestamos "C" (de crear) a la primera pregunta.

Se nos pregunta entonces por el número del concepto a crear y su descripción. A continuación las cantidades mensuales, todas ellas sin signo. El signo va implícito en el número de concepto, y por ello no es necesario. Terminada la introducción de las cantidades, para el año completo, vuelve a aparecer la pregunta de si se desea modificar —en este caso crear— nuevos conceptos. Cuando ya se han introducido todos los necesarios, se responde N y el programa entra en fase de cálculo.

Como este cálculo dura unos cuantos segundos, y para solicitar que seamos pacientes, aparece un aviso "CALCULANDO".




```

10 REM PREVISION DE GASTOS
11 REM
12 REM EL ORDENADOR PERSONAL M S,G
13 REM
14 REM 06.02.82
15 REM
90 D$ = CHR$(4):E$ = CHR$(27)
91 O$ = D$ + "OPEN VISI.D2"
92 C$ = D$ + "CLOSE VISI"
93 L$ = D$ + "READ VISI"
94 W$ = D$ + "WRITE VISI"
95 DE$ = D$ + "DELETE VISI"
100 DIM C$(12),T(16,12)
110 HOME
120 VTAB 4: PRINT "PREVISION ECONOMICA
ANUAL": VTAB 16
200 INPUT "DESEA CREAR O MODIFICAR (C,M)
":R$
210 IF R$ = "C" GOTO 300
220 IF R$ = "M" GOTO 200
250 GOSUB 1000
300 GOSUB 2000
350 HOME: INVERSE: VTAB 10: PRINT "
CALCULANDO " NORMAL
400 GOSUB 4000
500 GOSUB 5000
600 PRINT "SALVAR,MODIFICAR O FIN ? "
610 INPUT "":R$
620 IF R$ = "S" GOTO 700
630 IF R$ = "M" GOTO 800
640 IF R$ = "F" GOTO 900
650 GOTO 610
700 GOSUB 6000
800 GOSUB 2000
810 GOSUB 3000
820 GOTO 400
900 REM
990 END
991 REM
992 REM
993 REM
1000 REM
1001 REM CARGA DE LA TABLA
1002 REM -----
1003 REM
1100 PRINT O$: PRINT L$: REM ABRIR Y LE
ER
1200 FOR I = 1 TO 12
1210 INPUT C$(I)
1230 NEXT
1250 FOR I = 1 TO 12
1260 FOR M = 1 TO 12
1270 INPUT T(I,M)
1290 NEXT M: NEXT I
1300 PRINT C$
2000 REM
2001 REM MODIFICACION DE VALORES
2002 REM -----
2003 REM
2500 REM
2590 HOME
2600 PRINT "INGRESOS 1 A 6 GASTOS 7 A
12"
2602 PRINT: INPUT "DESEA MODIFICAR CON
CEPTOS ? ":R$: PRINT
2604 IF R$ = "N" THEN RETURN
2610 INPUT "Nº DE CONCEPTO ":I
2620 IF I > 12 THEN RETURN
2622 INPUT "CONCEPTO ":C$(I)
2625 PRINT "MES CANTIDAD"
2630 FOR M = 1 TO 12
2635 PRINT M;" ":
2640 INPUT " ":T(I,M)
2660 NEXT M
2680 GOTO 2590
3000 REM
3001 REM MODIFICACION SINGULAR
3002 REM -----
3003 REM
3100 PRINT "CONCEPTO,MES,CANTIDAD "
3110 INPUT I,M,T
3140 IF I > 12 THEN RETURN

```

```

3150 T(I,M) = T
3200 GOTO 3110
4000 REM
4001 REM CALCULO
4002 REM -----
4003 REM
4100 FOR M = 1 TO 12
4110 T(14,M) = 0:T(15,M) = 0
4120 FOR I = 1 TO 6
4130 T(14,M) = T(14,M) + T(I,M)
4140 T(15,M) = T(15,M) + T(I + 6,M)
4150 NEXT I
4200 T(16,M) = T(13,M) + T(14,M) - T(15,
M)
4210 IF M < 12 THEN T(13,M + 1) = T(16,
M)
4300 NEXT M
4310 RETURN
5000 REM
5001 REM IMPRESION
5002 REM -----
5003 REM
5090 SW = 0
5100 HOME:W = 4
5110 PRINT TAB(10);"B A L A N C E"
5120 PRINT:PRINT "Nº CONCEPTO
1 2 3 4 5 6"
5250 FOR J = 1 TO 12
5255 I = J
5260 IF C$(J) = "" GOTO 5330
5300 N$ = C$(J)
5320 GOSUB 7000
5330 NEXT J
5335 W = W + 2
5340 N$ = "SALDO ANT."
5345 I = 13
5350 GOSUB 7000
5360 N$ = "INGRESOS"
5365 I = 14
5370 GOSUB 7000
5380 N$ = "GASTOS"
5385 I = 15
5390 GOSUB 7000
5392 PRINT "-----"
5393 W = W + 1
5395 I = 16
5400 N$ = "SALDO MES"
5410 GOSUB 7000
5420 SW = SW + 6
5430 IF SW > 6 THEN RETURN
5435 PRINT
5440 INPUT "LISTO ? ":R$
5450 GOTO 5100
6000 REM
6001 REM ESCRITURA DE LA TABLA
6002 REM -----
6003 REM
6100 PRINT O$: PRINT DE$: PRINT O$: PRI
NT W$: REM ABRIR,BORRAR,ABRIR Y ESCRIBIR
6200 FOR I = 1 TO 12
6210 PRINT C$(I)
6220 NEXT I
6300 FOR I = 1 TO 12
6310 FOR M = 1 TO 12
6320 PRINT T(I,M)
6330 NEXT M: NEXT I
6400 PRINT C$
6410 RETURN
7000 REM
7001 REM IMPRESION LINEA
7003 REM
7080 W = W + 1
7090 VTAB W
7100 PRINT I;" ":N$
7105 HTAB 17: VTAB W
7110 FOR M = 1 + SW TO 6 + SW
7200 K$ = " " + STR$(T(I,M))
7210 PRINT RIGHT$(K$,4);
7250 NEXT M
7260 PRINT
7270 RETURN

```

A continuación, la visualización de los datos y resultados correspondientes al primer semestre. La limitación a seis meses, viene impuesta por el formato de pantalla, que en nuestro caso es de 40 posiciones x 24 líneas.

Con sólo pulsar RETURN, se obtiene el segundo semestre seguido de la opción SALVAR, MODIFICAR o FIN.

Por SALVAR, se entiende volcar a discos C\$ y T para uso futuro bajo el nombre VISI (en honor de VISICALC) y en la unidad D2. Previamente se hace un borrado de VISI. El programa termina.

Por FIN, el programa termina y lo simulado se pierde.

Por MODIFICAR, tenemos opción a modificar directamente cada casilla de T. Para ello hay que dar los datos: Número de concepto, mes, cantidad. Para salir del bucle, introducir 13,0,0 Return.

Modificaciones.

Como todos los problemas que aparezcan en el Ordenador Personal, sólo se pretende presentar la idea, una idea que funciona, pero que puede ser mejorada y embellecida. Mejorada en cuanto a prestación. Por ejemplo: si entre 400 y 500 se inserta pregunta sobre forma de impresión, se puede obtener la misma, tanto en pantalla como en impresora. En el papel no habrá la limitación de las 40 columnas. No hemos tratado el tema, porque una vez desarrollada la idea, el programa puede resultar el doble de largo, si se cuida la salida. El BASIC apple-soft tiene el defecto de carecer de PRINT USING y hay que hacer juegos malabares, para que las columnas de datos estén alineadas.

El organigrama se expone en la figura 2.

Introducción:
Christopher Disapean.
Programa: Miguel Solano.



ME HA DEJADO TIESO

COMPUTEC 5/1 **ORDENADOR PROFESIONAL**

- Z80A 4 MHz
- 64K RAM
- 16K ROM
- 2 DISQUETTES CADA UNO DE 350K FORMATEADOS
- TECLADO ERGONOMICO 97 TECLAS
- PANTALLA FOSFORO VERDE ORIENTABLE 12" ALTA RESOLUCION
- SALIDAS: 2 SERIES Y 1 PARALELO
- CP/M COMPATIBLE
- OPCION DISCO DURO 15 MB

COMELTA, S. A.

C/. Emilio Muñoz, 41
MADRID (17)
Teléf. 754 30 01
Telex: 42007 CETA-E

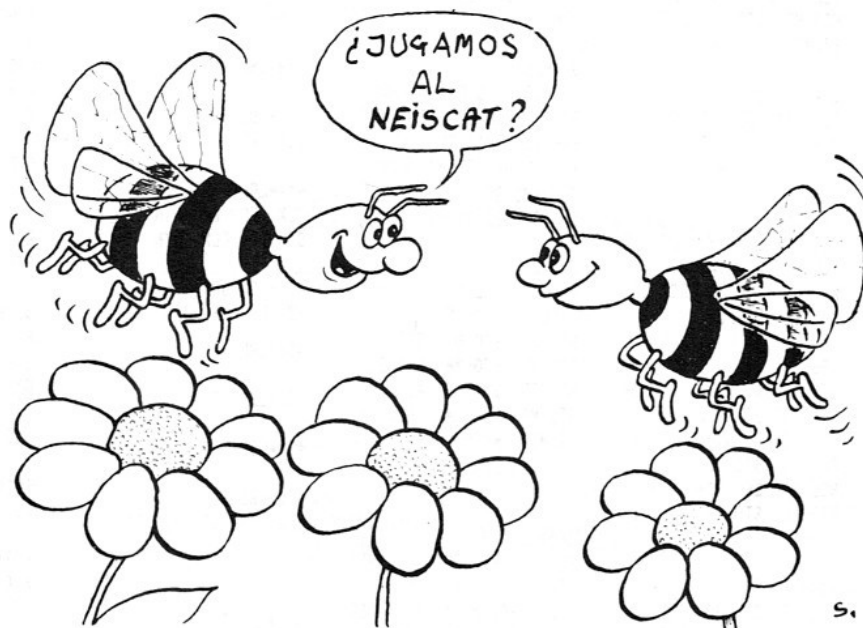
C/. Pedro IV, 84 - 5.º
Barcelona (5)
Telef. 300 77 12
Telex: 51934 CETA-E

C. T. E., S. A.

C/. Hernández Iglesias, 17
MADRID (27)
Telef. 267 52 36
Telex: 46781

M.A.C., S. A.

Avda. de la Industria, 52
Alcobendas (MADRID)
Tel. 653 36 23



El juego de Neiscat

No se extrañe si aún no conoce el juego de Neiscat: Acaba de ver la luz.

Sin embargo, la 41 CV ya lo conoce...

A partir de ahora, va a descubrir en su HP 41 CV un adversario de talla (pueden ser necesarios hasta 296 registros) y bastante hábil.

El Neiscat se juega entre dos, sobre un damero hexagonal (fig. 1) constituido de casillas también hexagonales: 19, 37 o 61 según escoja Vd. jugar en un damero de 3, 4 o 5 casillas de lado. Cada jugador dispone de un número ilimitado de fichas de doble cara. La cara superior lleva uno de los números 1, 2 o 3, y la cara inferior es del color del jugador (blanco o negro).

Los dos jugadores, alternativamente, van colocando en cualquier casilla libre del damero, una ficha (con la cara numerada visible) con el fin de realizar alineaciones "1-2-3" según cualquiera de las seis direcciones posibles (fig. 1).

Se puede "concluir" una línea, indistintamente, por una ficha núm. 1, 2 o 3 (1 si ya están colocados el 2 y el 3; 2 si va rodeado de 1 y 3...). Una ficha colocada pertenece al juego y por lo tanto, puede ser utilizada por **cada uno de los dos jugadores** para conseguir una línea.

Sin embargo, (¡Tenía que haber alguna excepción!), la ficha de conclusión de una serie, trátase de un 1, un 2 o un 3, tiene que ser colocada enseñando el color del jugador y quedar así neutralizada: **ya no puede intervenir en la realización de otra línea.** La puntuación del jugador es igual al valor del número que lleve la ficha de conclusión de la serie. Y aunque se concluyan varias series a la vez, sólo se contabiliza una.

No haga el juego del adversario

Puede parecer fácil realizar estas líneas, pero nada de eso. Puesto que, **¡El número de la ficha a colocar le es impuesto!**

En efecto, los dos adversarios deben jugar en el orden de las fichas numeradas 1, 2, 3, 1, 2, 3... Es decir que los jugadores A y B colocarán sucesivamente:

Jugador A	1	3	2	...
Jugador B	2	1	3	...

y la dificultad reside en que una ficha colocada puede hacer inmediatamente el juego del adversario.

Otro detalle que conviene conocer: Al principio del juego, la casilla central está ocupada por una ficha de valor 3 que podrá servir de base para la constitución de series ganadoras. Por lo tanto, al principio de la partida siempre habrá un número de

casillas libres múltiplo de 6, lo cual equilibra de manera óptima las posibilidades de los dos jugadores. Termina la partida cuando ya no queda ninguna casilla libre o cuando resulta evidente que ya no se puede realizar ninguna línea.

Una vez introducido el programa en la HP 41, hay que completarlo por la rutina ALEA o cualquier otra que genere un número pseudoaleatorio. Este es lanzado por XEQ "NSCAT" y es pedida la dimensión del damero de juego: de 1 a 5 casillas por lado. Los

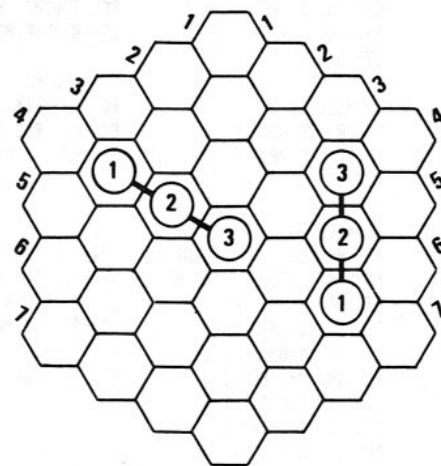


Fig. 1
Dimensión 4: plan del juego con dos ejemplos de series terminadas.

```

PRP "NSCAT" 165*LBL 13
CLX STO 20 STO 21
FC? 03 GTO 37 STO 03
24.021 STO 06

01*LBL "NSCAT"
*VSC* CLRG CF 00 4
*DIM? <6* PROMPT
STO 14 STO 05 CHS
11.011 + STO 06
30.029 RCL 14 + LASTX
1 E3 / + STO 07

22*LBL 00
RCL 06 ST+ 07 1

26*LBL 01
STO IND 07 ISG 07
GTO 01 DSE 06 RDN
DSE 05 GTO 00 .999
ST+ 06 RCL 14 1 -
STO 05

40*LBL 11
RCL 06 ST+ 07 1

44*LBL 12
STO IND 07 ISG 07
GTO 12 ISG 06 DSE 05
GTO 11 10 STO 09 CHS
STO 12 1 STO 10
STO 00 CHS STO 11 11
STO 08 ST- 13 RCL 14
* 30 + 15 STO IND Y
SF 01 CF 02 CF 03
CF 29 SF 07 8.013
STO 28 CLX "PREP"
TONE 5

79*LBL 15
FIX 0 CF 05 CF 06
CF 22 PROMPT FS? 22
GTO 02 FS? 07 GTO 16
CLX STO 20 STO 21
CF IND 00 RCL 00 3
DSE Y RDN STO 00
SF IND 00 RCL 16
X<> 17 STO 16 GTO 10

103*LBL 02
X=0? GTO 28 30 +
STO 01 RCL IND 01
RCL 10 "ERR:" X=Y?
GTO 15 2 RCL 00 Y1X
RND RCL 10 - ST+ X
ST+ IND 01

122*LBL 10
SF 00 CF 05 CF 06
CF 07 CF 08 RCL 28
STO 07

130*LBL 04
RCL 00 30 + XEQ IND X
FS?C 05 GTO 27 ISG 07
GTO 04 4 FS? 02 3
FS? 03 5 STO 06
RCL 20 X=0? GTO 13
RCL IND 20 RCL 10 X=Y?
GTO 06 RCL 21 X=0?
GTO 13 RCL IND 21
RCL 10 X=Y? GTO 13

159*LBL 06
RCL 2 STO IND 06 3
ST+ 06 SF IND 06

316*LBL 09
RCL 2 STO 04 SF 07
RTN

321*LBL 33
XEQ 17 STO 02 4 X=Y?
SF 05 RCL 02 -2 X=Y?
RTN RCL 01 RCL IND 07
+ STO 04 SF 07 RTN
337*LBL 26
CF 00 XEQ 25

340*LBL 16
RCL 14 RCL 08 * 19 +
FS?C 07 GTO 23 CLX
STO 03 STO 04 STO 18
STO 19 STO 20 STO 21
STO 25 STO 26 41
STO 01 RCL 14 .022 *
41.019 + STO 06

365*LBL 20
RCL IND 01 RCL 10 X=Y?
GTO 21 RCL 01 30 -
CLA ARCL X "I?" AVIEW
CLX STO 02 RCL 28
STO 07

381*LBL 17
RCL 01 RCL IND 07 +
ST+ L RCL IND X X=0?
GTO 07 RCL IND L -

391*LBL 07
FS? 00 RTN 53 +
XEQ IND X ST+ 02
ISG 07 GTO 17 FS? 03
GTO 22 8.01 STO 07

404*LBL 18
RCL 01 RCL IND 07 +
RCL 01 LASTX -
RCL IND Y RCL IND Y +
FS? 00 RTN 69 +
XEQ IND X ST+ 02
ISG 07 GTO 18 GTO 22

423*LBL 53
CLX RTN

426*LBL 54

427*LBL 69

428*LBL 56

429*LBL 68

430*LBL 70

431*LBL 72

432*LBL 84

433*LBL 55

434*LBL 49

435*LBL 41

436*LBL 67

437*LBL 65

438*LBL 61

439*LBL 75

440*LBL 91

441*LBL 99
CLX RTN

444*LBL 71
CLX FC? 02 RTN RCL 01
STO 25 CLX RTN

452*LBL 60
-2 FS? 03 CLX RTN

457*LBL 47

458*LBL 77

459*LBL 79

460*LBL 83
CLX FS? 02 -2 RTN

465*LBL 51
CLX FC? 03 RTN RCL 01
RCL IND 07 + RCL 18
X=Y? STO 19 RDN
STO 18 RCL 09 RTN

479*LBL 39
CLX FS? 01 -5 RTN

484*LBL 59
-6 FS? 03 -2 FC? 01
RTN RCL IND 07 ST+ X
RCL 01 + RCL 18 X=Y?
STO 19 RDN STO 18
RCL 09 RTN

501*LBL 57
RCL 09 ST+ X FS? 01
RCL 12 FS? 02 30 RTN

509*LBL 45
2 FS? 02 -6 FS? 03
RCL 10 RTN

516*LBL 73
CLX FS? 02 -15 RTN

521*LBL 85
CLX FC? 02 RTN
RCL IND 07 RCL 01 +
RCL IND X RCL 10 -
X=0? GTO 03 RCL 01
RCL IND 07 - R1

537*LBL 03
RDN RCL 18 X=Y?
STO 19 RDN STO 18 2
RTN

546*LBL 87
20 FS? 01 15 RTN

551*LBL 76
CLX FS? 02 -2 RTN

556*LBL 22
RCL 01 RCL 04 RCL 03
X=0? GTO 14 RDN
RCL 02 X<Y? GTO 21
STO 04 X=Y? GTO 14
RDN XEQ "ALEA" .3 -
X=0? GTO 21

575*LBL 14
RCL 2 RCL 25 X=Y? CLX
STO 26 RDN STO 03
RCL 18 STO 20 RCL 19
STO 21

587*LBL 21
RCL 18 ST+ 01 CLX
STO 18 STO 19 ISG 06
GTO 20 RCL 26 X=0?
GTO 02 X<> 22 X<> 23
STO 24

601*LBL 02
RCL 03

603*LBL 23
X=0? GTO 28 STO 01
SF 00 RCL 28 STO 07

610*LBL 38
FS? 02 XEQ 18 FC? 02
XEQ 17 FS? 01 -8
FS? 02 18 FS? 03 4
X=Y? GTO 24 ISG 07
GTO 38 CF 00 2 RCL 00
Y1X RND RCL 10 -
ST+ X ST+ IND 01 CLA
ARCL 00 "I:" XEQ 25
RCL 01 30 - ARCL X
TONE 5 GTO 15

644*LBL 24
CF 00 RCL 00 ST+ 17
CF IND 00 CLX STO 20
STO 21 STO IND 01 CLA
ARCL 00 "I:" XEQ 25
RCL 01 30 - ARCL X
XEQ D TONE 9 TONE 8
GTO 15

665*LBL 25
CF IND 00 RCL 00
RCL 10 + 4 MOD X=0?
RCL 10 STO 00
SF IND 00 RTN

677*LBL 27
CF 06 CF 07 CF 08
FC? 22 GTO 26 RCL 00
ST+ 16 CLX STO IND 01
RDN "OK" XEQ D TONE 0
PSE PSE CLD GTO 26

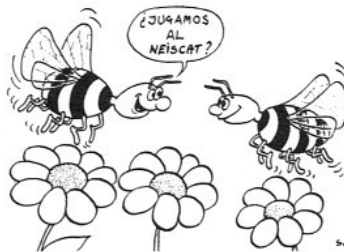
695*LBL 28
CF IND 00 "FIN" XEQ D
TONE 5 STOP
GTO "NSCAT"

702*LBL D
"I S=" ARCL 16 "I--"
ARCL 17 AVIEW END

PRP "ALEA"

01*LBL "ALEA"
RCL 29 PI + 5 Y1X
FRC STO 29 END

```

El juego de Neiscat

Líneas del programa

1 a 78	Iniciación
79 a 335	Análisis del golpe contrario
336 a 601	Análisis del juego
602 a 707	Juego de una casilla



valores 1 y 2 son triviales; el 3 permite "entrenarse"; y las dimensiones 4 y 5 son las más interesantes. A falta de esta introducción, el programa escoge la dimensión 4.

Una vez establecido su plan de juego, el programa se detiene en el mensaje "PREP.": ¡A sus fichas! Si debe empezar la HP, pulsar R/S. Si no, entre Vd. las coordenadas de la casilla que decide jugar antes de pulsar R/S. Recuerde que coloca una ficha de un determinado valor: a lo largo de toda la partida, la HP indicará dicho número por medio de los indicadores binarios 1, 2 y 3 (banderas).

Vd. no puede "hacer trampas", pues el mensaje "ERR" indicaría que juega una casilla ya ocupada. En cambio, puede hacer que la HP juegue en su lugar o que sencillamente juegue contra sí misma pulsando R/S sin introducir coordenadas de casilla.

¿Cómo determinar las coordenadas de una casilla? Explicamos el método en

la figura 2: Aquí la casilla central es la 44, la 33 se encuentra inmediatamente encima...

Una vez memorizado su golpe, la HP analizará sus consecuencias (bandera 0), visualizará eventualmente un resultado (puntuación) y, por último, se pondrá a estudiar su respuesta. En algunos casos, ésta será inmediata, pero, en general, examinará todo el plan de juego, casilla por casilla. Para que lo lleve Vd. con paciencia, le indica las coordenadas de las casillas examinadas: "XY?".

El tiempo de respuesta global es, en el caso más desfavorable, de 3 minutos en dimensión 3, 6 minutos en 5 y 10 minutos en 5. Pero, se lo repetimos, por término medio, los golpes son más rápidos.

El juego de la máquina es indicado por un bip y visualizado: "nº de ficha: nº de casilla". Por ejemplo, "2:35" significa que la HP coloca una ficha nº 2 en la casilla 35. Al mismo tiempo presenta el indicador de número de la ficha (flag) que debe

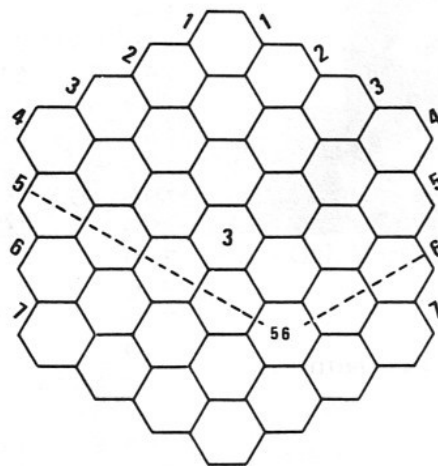


Fig. 2
Las dos direcciones oblicuas descendientes, definen las coordenadas de cada casilla.

El programa

Por lo que se refiere al programa, ocupa 174 registros y hay que añadirle una rutina que genere un número pseudoaleatorio ("ALEA") comprendido entre 0 y 1. Son necesarios numerosos registros de memoria: 97 en dimensión 3, 119 en 4 y 141 en 5; pero la mayoría de ellos permanecen... inutilizados (éste es el caso de 52 registros en dimensión 4). Cerca del 60 % de desecho como promedio, ¿No resulta horrible? La desventaja de una gestión más puntillosa de las memorias sería un despilfarro del tiempo de análisis. Si le apetece...

La codificación de las casillas está hecha de la siguiente forma: 0 (fuera de juego, 1(casilla libre), 3 (nº 1), 7 (nº 2) y 15 (nº 3), según la fórmula $2^{(nº \text{ de ficha} + 1)} - 1$. La casilla número XY corresponde al registro nº $30 + XY$. Por último, señalemos que el corazón del programa, ahí donde se elabora la estrategia, se encuentra en las líneas 422 a 554.

ahora colocar Vd. Para jugar, entre sus coordenadas seguidas de R/S.

Toda partida debe terminarse

Una serie concluida es anunciada por una musiquilla y el mensaje "S = s1 - s2" donde s1 es su puntuación y s2 la de la HP. Si hizo jugar la HP en su lugar, las puntuaciones resultan invertidas. La conclusión de una de sus series es anunciada por un mensaje semejante pero precedido de "OK".

Al final de una partida, la HP detecta por sí misma que no queda ninguna casilla libre, pero, en cambio, cuando ya no se puede realizar ninguna serie, hay que indicárselo. Para ello, tiene Vd. que introducir 0 en vez de las coordenadas de una casilla: el mensaje "FIN S = ...:..." da el tanteo de final de partida.

Robert Pulluad

GRAFICOS EN EL OSBORNE 1



Aunque el Osborne 1 no dispone de verdaderas capacidades gráficas, es posible aprovechar el bloque de caracteres gráficos predefinido para dibujar algunas cosas interesantes. Vemos cómo puede hacerse a través del siguiente texto y de algunos ejemplos.

La resolución alfanumérica de la pantalla del Osborne 1 es de 32 líneas de 128 posiciones cada una, lo que supone que, en total, disponemos de 4.096 posiciones, de las

cuales sólo son visibles al mismo tiempo 1.248 (ventana de 24 por 52). Con fines gráficos una serie de caracteres especiales pueden ser visualizados en cada una de las

4.096 posiciones por medio de las sentencias POKE y PRINT CHR\$.

La sentencia POKE es una de las más peligrosas que pueden usarse en un programa Basic, pues nos permite situar valores en posiciones determinadas de la memoria de nuestro microordenador, con todos los problemas que puede acarrear el introducir algo en una posición errónea. El formato de la sentencia POKE es:

POKE dirección, valor
en la que "dirección" tiene que estar comprendida entre 0 y

```
100 '
110 '      PROGRAMA -2-
120 '
130 PRINT CHR$(26): PRINT TAB(60)
140 K = 127
150 D = 0
160 P = 63002!
170 POKE P,K
180 FOR N = 1 TO 25
190   IF D = 4 THEN D = 0
200   D = D + 1
210   IF D = 1 THEN I = 128
220   IF D = 2 THEN I = 1
230   IF D = 3 THEN I = -128
240   IF D = 4 THEN I = -1
250   FOR S = 1 TO N
260     P = P + I
270     POKE P,K
280     IF I = 128 OR I = -128 THEN GOTO 310
290     P = P + I
300     POKE P,K
310   NEXT S
320 NEXT N
330 IF K = 32 GOTO 140
340 K = 32
350 GOTO 150
```

```
100 '
110 '      PROGRAMA -1-
120 '
130 PRINT CHR$(26): PRINT TAB(60)
140 P = 61440!
150 A = 0
160 B = 31
170 K = 0
180 FOR I = A TO B
190   IF K > 45 THEN GOTO 220
200   K = 0
210   P = P + 256
220   POKE P + K, I
230   K = K + 5
240 NEXT I
250 IF A = 127 THEN GOTO 250
260 A = 127
270 B = 160
280 P = 63360!
290 GOTO 170
```



```

100 ?
110 ?      PROGRAMA -3-
120 ?
130 PRINT CHR$(26)
140 P = 62976!
150 IF P > 61440! AND P < 65535! THEN POKE P,127
160 C$ = INKEY$
170 IF C$ = "1" THEN P = P + 127: GOTO 150
180 IF C$ = "2" THEN P = P + 128: GOTO 150
190 IF C$ = "3" THEN P = P + 129: GOTO 150
200 IF C$ = "4" THEN P = P - 1: GOTO 150
210 IF C$ = "6" THEN P = P + 1: GOTO 150
220 IF C$ = "7" THEN P = P - 129: GOTO 150
230 IF C$ = "8" THEN P = P - 128: GOTO 150
240 IF C$ = "9" THEN P = P - 127: GOTO 150
250 GOTO 160

```

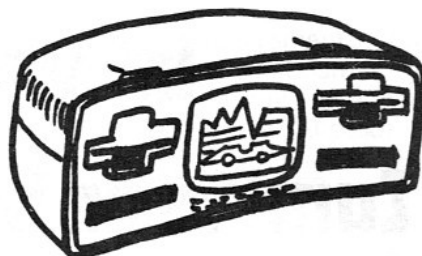
65.535, y "valor" es un octeto entre 0 y 255. Así, la sentencia POKE 64000,65 colocaría en la posición de memoria 64000 el código ASCII 65 que corresponde a la letra "A".

La memoria video del Osborne 1 comienza en la posición 61440 y termina en la 65535, correspondiendo la primera de ellas a la esquina superior izquierda de la pantalla, y la segunda a la inferior derecha. Los caracteres gráficos disponibles pueden obtenerse en

la pantalla con la ejecución del programa 1.

Con anterioridad indicamos que los dos medios de obtener caracteres gráficos eran las sentencias POKE y PRINT CHR\$. No obstante esta última no puede ser empleada ya que no permite direccionar una posición cualquiera de la pantalla, por lo que no queda más remedio, en principio, que usar la "peligrosa" POKE.

Como una imagen vale más que mil palabras les animo a que

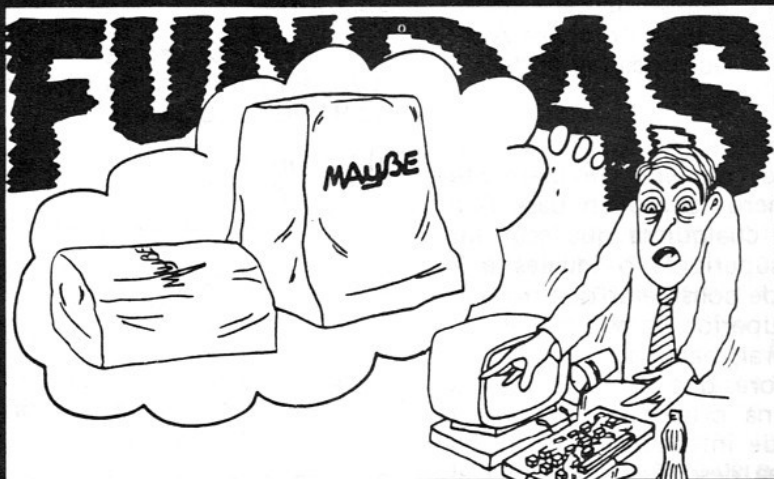


prueben los programas números 2 y 3. El 2 dibuja indefinidamente una espiral en la parte visible de la pantalla, y, el 3, le permite realizar el gráfico que usted desee empleando el bloque de teclas numéricas situado a la derecha de su teclado como cursores de dirección.

Tenga también en cuenta que con objeto de no estropear las imágenes que aparecen en la pantalla, ninguno de los tres programas termina normalmente, por lo que deberá teclear CONTROL C si quiere detenerlos.

en realidad la realización de gráficos en el Osborne 1 no es demasiado compleja y sus posibilidades son bastante interesantes. Con un poquito de imaginación, y de paciencia, pueden crearse buenos gráficos de barras y, por qué no, divertidos juegos.

Luis de Cáceres Muñoz



MAYBE

BARCELONA - 6 - Brusi, 102 - Entresuelo 3º
Tfno. (93) - 201 21 03.

MADRID - 10 - Gal. Martínez Campos, 5 - Bajo izqda.
Tfno. (91) - 445 84 38 - 446 60 18.

MAYBE
le evitará
encontrarse
en esta
situación

BOLETIN DE PEDIDO a mandar a MAYBE - Gal. Martínez Campos, 5 - Bajo izqda. Madrid - 10

Les ruego me manden :

..... fundas (1) para Apple II con monitor de 9"	a 1.200Pts :
..... fundas para Apple II con monitor NEC de 12"	a 1.250Pts :
..... fundas para Apple II con monitor Philips de 12"	a 1.250Pts :
..... fundas para Apple III sin profile.	a 1.250Pts :
..... fundas para impresora de 80 columnas.	a 700Pts :
..... fundas para impresora de 132 columnas.	a 750Pts :

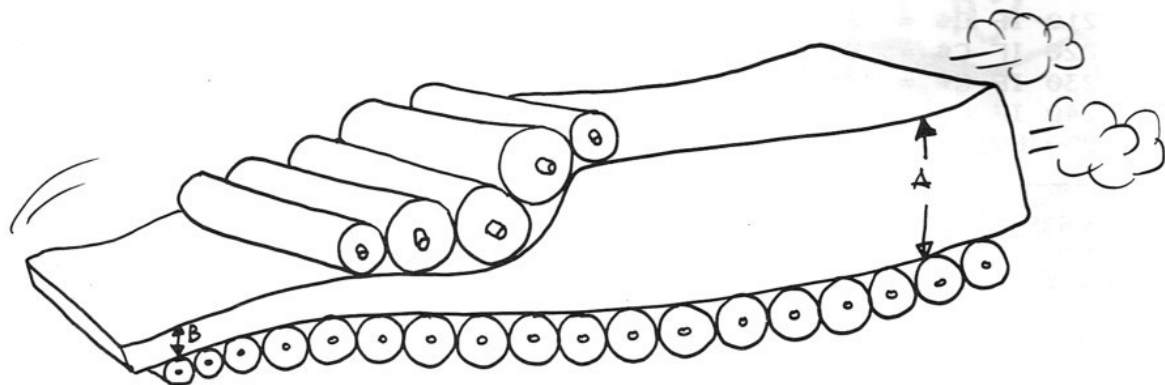
pagando con ☐ talón adjunto o ☐ contra reembolso el Total de (añadir gastos de envío: 250 pts. por 2 fundas y 175 pts. por 1 funda).

Nombre : Apellidos :
 Empresa : Cargo :
 Calle : No. : Tel. :
 Ciudad : D.P. : Provincia :
 Fecha : Firma :

(1) Apple con dos drives.

Informàtica de bolsillo

Cambie de base cuando lo necesite. Con este programa para calculadora HP.



Aunque es verdad que se pueden hacer "a priori" más cosas utilizando "disquettes" en vez de "cassettes", no es menos cierto que con frecuencia la creatividad hace aumentar el campo de utilización de los ordenadores. Este es un programa cuyas aplicaciones rebasan el campo neto de posibilidades de las máquinas que nosotros utilizamos, ya sean calculadoras de bolsillo u ordenadores personales. El objeto de este programa es en apariencia una operación muy conocida por todos: el cambio de base y la calculadora utilizada es una HP 25 C (o cualquier HP que tenga como mínimo 49 pasos).

No hay propietario de calculadora programable que no haya "metido en su máquina", alguna vez, un programa de cambio de base A a base B.

Los programas que se suministran con las calculadoras programables, en este caso la HP 25 C, se limitan generalmente a transformar un número de base A en base 10 y recíprocamente de base 10 en base B.

Primera ventaja de este programa: que convierte directamente un número fuente en base A a base B, cualquiera que sean las bases, superiores o iguales a 2 (se puede considerar que no hay límite superior, ya que se trabaja, en general, bastante poco en bases mayores que 10^{10}). Esto evita hacer una primera conversión y tener que introducir un segundo programa después para acabar el trabajo.

La segunda ventaja muy fundamental (¡y qué importante es!) reside en el hecho de que en el caso de transformaciones de números expresados en bases grandes a números expresados en bases pequeñas, la máquina no puede, en todos los casos, proporcionar todas las cifras. Generalmente da un resultado en forma de número decimal acompañado de una potencia de 10 y el utilizador se encuentra en la más

desesperante imposibilidad de conocer las últimas cifras de su resultado.

Veamos dos ejemplos, donde se recogen los resultados correspondientes de un programa normal que da una respuesta global y el obtenido con el programa adjunto.

1.— ¿Qué número es el equivalente en base 4 al número $X = 92\ 91\ 90\ 89$ (base 93)?

a) Un programa normal da el resultado $X = 1,0131110 \cdot 10^{13}$ (base 4). Un resultado muy pobre.

b) Con el programa cuyo listado aparece en este artículo
 $X = 10131110311022$ (base 4)

Se obtienen, como se ve, las 6 cifras en cursiva que faltaban en la primera respuesta.

2.— ¿Cuál es el equivalente en base 2 del número $Y = 18\ 18\ 18$ (base 19)?

a) Resultado normal:
 $Y = 1,1010110 \cdot 10^{12}$ (base 2)
se ve que faltan cinco cifras.

b) Resultado "cifra a cifra":
 $Y = 1101011001010$ (base 2)

El programa tiene sus limitaciones de utilización, pero presenta la ventaja de llevar mucho más lejos los límites normales.

Veamos ahora cómo la máquina puede proporcionar resultados

Listado del programa

01 STO 1	25 RCL 5
02 $X \geq Y$	26 $GX \neq 0$
03 STO 0	27 GTO 13
04 R/S	28 RCL 3
05 STO 5	29 STO 5
06 1	30 RCL 1
07 STO 2	31 $FX < Y$
08 CLX	32 GTO 35
09 STO 3	33 $FX \neq Y$
10 R/S	34 GTO 48
11 $G 10^X$	35 :
12 STO 4	36 F INT
13 RCL 4	37 STO 2
14 $STO \div 5$	38 RCL 1
15 RCL 5	39 *
16 G FRAC	40 CHS
17 $STO - 5$	41 RCL 5
18 RCL 4	42 +
19 X	43 $G X < 0$
20 RCL 2	44 CLX
21 X	45 F PAUSE
22 $STO + 3$	46 RCL 2
23 RCL 0	47 GTO 29
24 $STO * 2$	48 RCL 5
	49 GTO 00

de 14 y 13 cifras en los dos ejemplos precedentes y muchos más en casos mucho más impresionantes, aunque su pantalla está limitada a 10 cifras. Para realizar esto, va a dar el resultado "cifra a cifra" y empezando por la derecha.

Ejemplo: en el caso 1.b) la máquina va a visualizar sucesivamente 2,2,0,1,1,3,0,1,1,1,3,1,0 y 1. Pero, atención, la máquina no es idiota, ella da siempre sus "cifras" separadamente. Sin embargo, el número de estas cifras sacadas simultáneamente en pantalla depende de la base final. La condición "n cifras por n cifras" se puede escribir

$$10^{n-1} < B \leq 10^n$$

En efecto, hagamos la conversión siguiente:

$$\begin{aligned} Z &= 66\ 65\ 64\ 63 \text{ (base 67)} \\ &= ??\ ??\ ??\ \dots \text{ (base 15)} \end{aligned}$$

El resultado de la máquina va a ser:

$$Z = 1\ 11\ 07\ 14\ 04\ 14\ 09 \text{ (base 15)}$$

pero lo habrá dado de la siguiente forma:

$$9,14,4,14,7,11 \text{ y } 1$$

No hay que olvidar los "ceros" que faltan, pues si la máquina visualiza 7, hay que interpretarlo como 07. En realidad, todo lo que hay que saber es que en base 15, 14 no es un número sino una cifra que corresponde a la letra griega "epsilon".

La máquina da automáticamente el formato del resultado, pero no sucede desgraciadamente lo mismo con el número introducido. En efecto, debido a la falta de capacidad de memoria, es preciso dar como información a la máquina, además de los datos de base de partida y llegada y el número fuente, el número de "cifras" (en salida) que corresponde a cada cifra, del número inicial. La condición relativa a n es la misma que antes, sin más que reemplazar B por A. No hay que olvidar que con una base de partida superior a 10, es preciso reservar por lo menos 2 "cifras" (en salida) por cada una.

O sea, si se quiere convertir 113,25,3 (base 157) en base N hay que dar a la máquina un número escrito así:

$$113025003$$

Se observa, además, a la vista del programa, que no utiliza ni

subprograma, ni decremento con bifurcación, ni direccionamiento indirecto y que puede funcionar con cualquier HP con 49 pasos ó más de programa o con cualquier "Texas" que tenga 100 pasos, aproximadamente. Es, por lo tanto, utilizable por una amplia gama de máquinas (sin contar con los ordenadores BASIC).

Para acabar, veamos el "modus operandi" del programa:

Se inicializa: 0 decimal, inicialización (F prgm o RST).

Se introduce: A (base de partida), después B (base de llegada), o sea A Enter B.

Se teclea N (el dato a tratar), sin olvidar "los ceros" no significativos y se pone en marcha, o sea: N (Base A) R/S.

La máquina se detiene de nuevo mostrando 0 (cero).

Se escribe ahora el número de "cifras" necesarias (en la salida) por cada cifra el número N que se acaba de introducir (con objeto de que la máquina pueda trabajar "cifra" a "cifra").

Se vuelve a poner en marcha por última vez con la operación:

Número de "cifras" R/S

Una vez iniciado el proceso, no hay más que ir a toda velocidad a buscar lápiz y papel y se puede ya anotar lo visualizado en pantalla, a partir de la derecha. Pero una vez más atención a los "ceros"!

Puede suceder que algunos lectores interesados, divertidos analizando el programa, no vean la utilidad de los pasos 43 y 44. En efecto ¿por qué plantear un problema en el caso de que se introduzca una "cifra" negativa? Desde luego, esto ocurre rarísimamente. Los curiosos pueden reemplazar al paso 44 por una instrucción inoperante (g NOP), después de hacer la conversión de base 99 en base 2 del número 98 98 98 98 y que intenten comprender porqué los - 1 son realmente ceros (¡mucha suerte!).

Una vez más se ve que las posibilidades de las máquinas no están en proporción directa con la inversión que representan. □

Olivier Barthélémy.

Juego de las siete y media para la CASIO FX-702-P

Partiré de la suposición de que todos (o la mayoría) sabeis jugar a las siete y media. El que no sepa o no recuerde bien las reglas, podrá aprender, o refrescarlas, mientras sigue el funcionamiento del programa o lo ejecuta en su máquina.

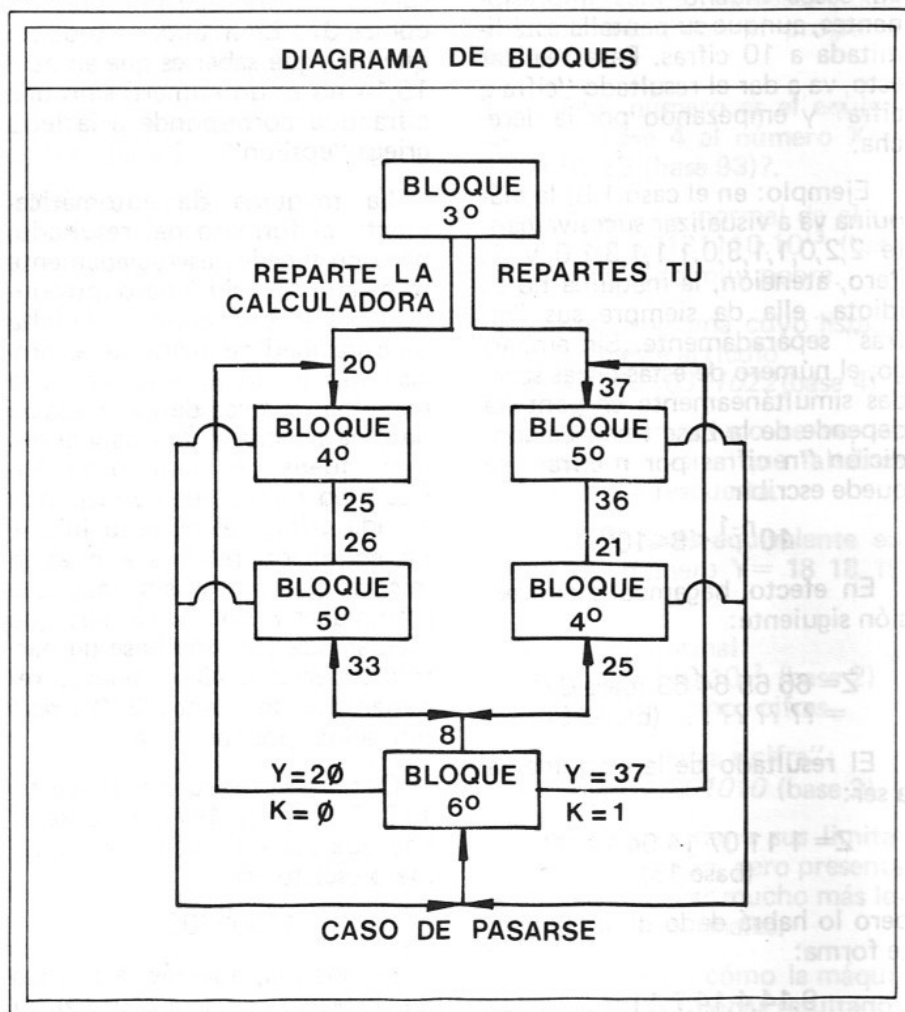
El programa en cuestión ocupa 950 pasos, (aunque cuando lo creé, recién conseguida la calculadora, ocupaba 1350) más 80 pasos resultantes de definir diez memorias extra para la matriz columna que actuará como baraja (que es española, de 40 cartas).

Este puede considerarse formado por seis bloques, prácticamente independientes, entre los cuales va saltando el control del programa. Cada bloque cumple una misión específica y, aunque pueda parecer que tiende a hacer más lento el programa, en realidad ocurre al revés. Los bloques en cuestión son los siguientes:

BLOQUE 1º.- Este tiene como misión el comprobar el estado de la baraja (contar las cartas que van quedando) y la reposición de la misma, cuando se llega al mínimo de cartas permitido. Está compuesto por las líneas 15 (que realiza el primer punto) y 16 (que se encarga de barajar).

Esta parte se ejecuta al principio del programa (para crear la baraja) y después de cada mano (comprobación y, en caso necesario, reposición).

BLOQUE 2º.- Es el encargado de generar la carta que nos sale cuando pedimos. Está compuesto por las líneas 2 a 7 inclusives. Además de generar carta, ad-



judica a esta un valor (a cada una el suyo menos las figuras que valen medio punto). En la baraja que usa la CASIO, la sota, el caballo y el rey están sustituidas por sus equivalentes de la baraja francesa, esto es, la "J" (Jack), la "Q" (Queen), y la "K" (King), respectivamente.

La 2ª línea se encarga de lo que propiamente es sacar la carta. La línea 3 comprueba que dicha carta está en la baraja. La 4 quita el naipe del juego y, en caso de no ser figura, dárselo al jugador. Las líneas restantes sirven para identificar las figuras.

BLOQUE 3º. Es el inicio - presentación del programa y está compuesto por las líneas 1 y 19. Su misión es hacer una limpieza (o inicialización) de las variables (comando VAC), indicar el tiempo de impresión de cada cartas y preguntarte (muy amable por su parte) si quieres empezar repartiendo tú.

En las líneas 19 y 24, aparece la instrucción IF A\$ > "O". Es la comprobación de tu respuesta a la pregunta que te ha hecho (en la línea 19 si quieres repartir y en la 24 si quieres carta). Esto es debido a que si este verano ligas con una extranjera, tal vez le sea más fácil, para decir "sí", escribir "yes" o "oui", pudiendo contestar también tú, de forma más rápida, con una simple "s" (esto lo leí en un número anterior del O.P.). El "no", en otros idiomas, es más estandar y si pones una simple "n", también te lo admite.

BLOQUE 4º. Es tu turno de juego, bien porque seas mano (empieza en la línea 20), bien porque, siendo mano la calculadora, ésta haya decidido plantarse, en cuyo caso el control pasa de la línea 36 (del 5º bloque) a la 21.

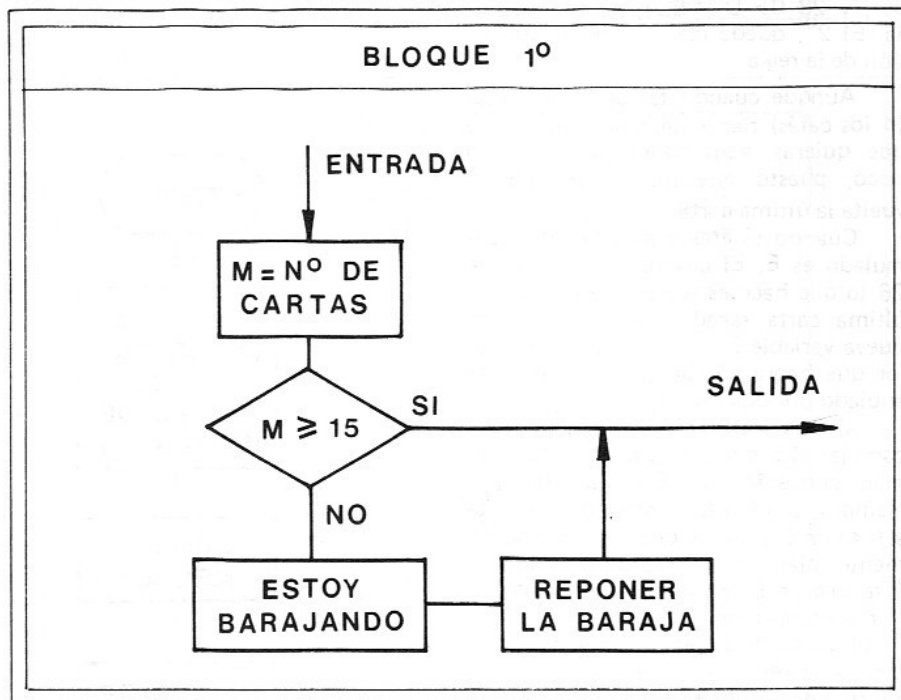
Este bloque comprende las líneas 20 a 25 siendo su misión: darte carta (para lo cual recurre al 2º bloque), comprobar si te pasas (lo cual es hecho, al igual que lo anterior en la línea 21), decirte el total de puntos que llevas acumulados (y la carta que te ha salido, todo en las líneas 22 y 24) y preguntarte si quieres otra carta.

Cuando te plantas, el control es transferido a la línea 26 (bloque 5º) en el caso de que seas mano (K=0). En el caso de que la calculadora sea mano, el control irá al bloque 6º (línea 8) donde se decidirá el ganador.

Si te has pasado, se pasa de la línea 21 a la 17 (bloque 6º) donde se dará como ganadora a la máquina.

BLOQUE 5º. Es el turno de juego del calculador y, básicamente un bloque análogo al anterior.

Cuando tu CASIO es mano, se entra al bloque a través de la línea 37 (donde se nos avisa que ella es mano y va a efectuar su baza). En el caso de que seas tú quien juegue primero, el control será transferido, una vez que te hayas plantado, de la línea 25 del bloque antes explicado a la 26, que es la

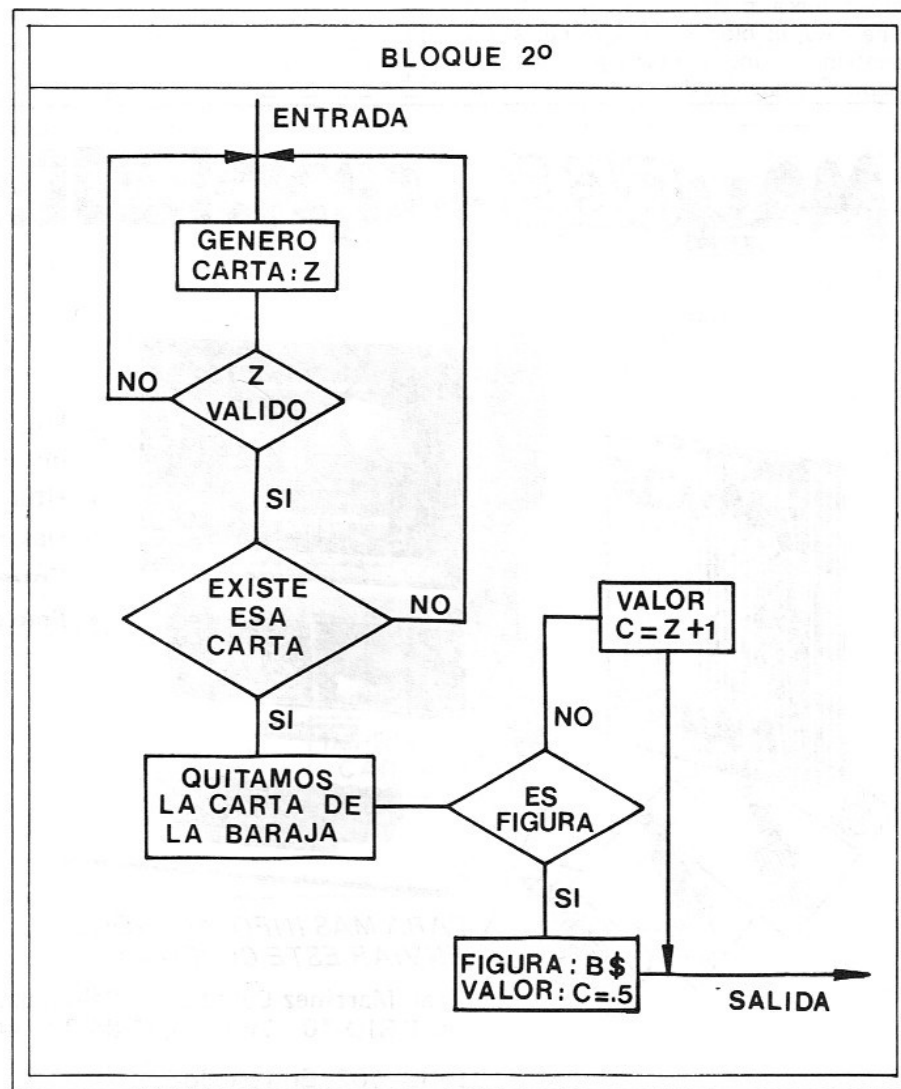


entrada, en este caso, del bloque que estamos estudiando ahora. Aquí, tengo que hacer un alto para explicar dos cosas: una regla básica del juego y su aplicación al programa.

La regla consiste en que el jugador que es mano tiene derecho cuando se

planta, a que una de sus cartas esté vuelta (oculta). Esto no es aplicable cuando el jugador se pasa o cuando se planta con siete y media, ya que en ambos casos debe avisar a su o sus contrarios.

El primer caso ya está recogido en las líneas 21 (IF E ≥ 8...), para el juga-



dor, y 28 (IF D $\geq$ 8...), para la máquina. El 2º, queda claro al ver la aplicación de la regla.

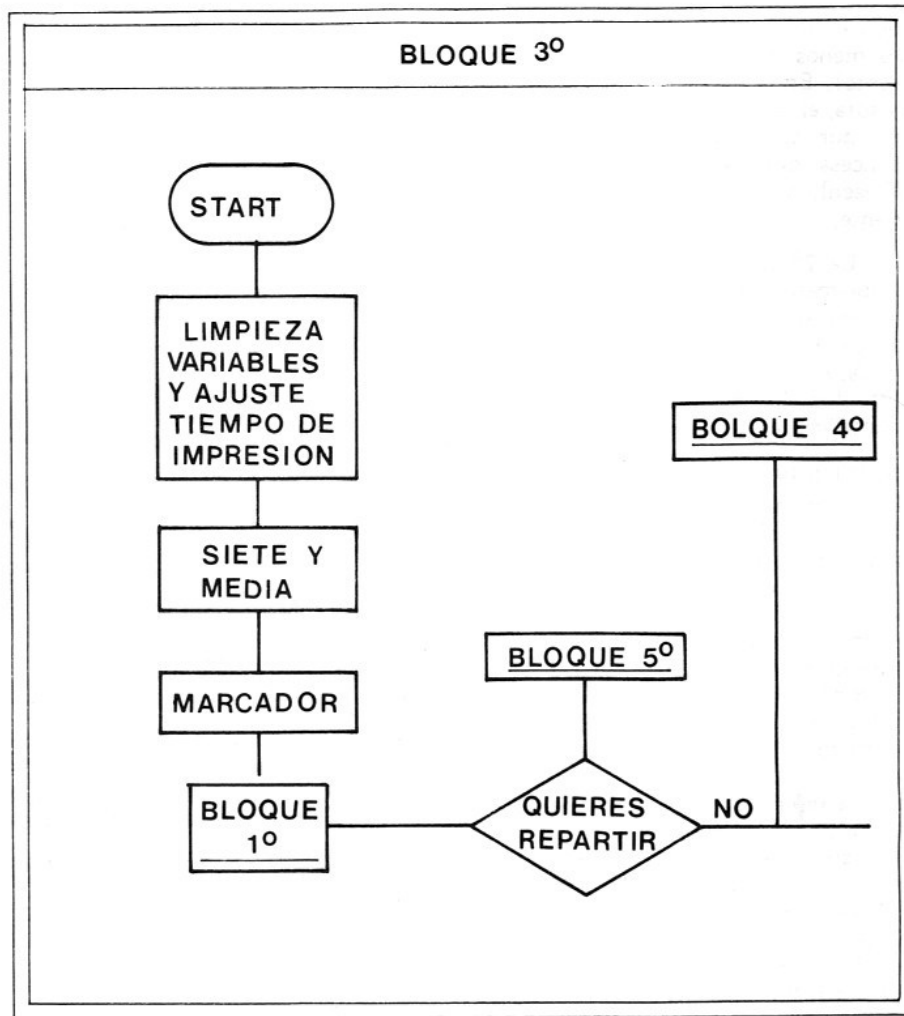
Aunque cuando te juegas las pelotas (o los cafés) tienes derecho a ocultar la que quieras, aquí habrás de ceder un poco, puesto que sólo se mantendrá vuelta la última carta.

Cuando tú eres mano, tu total acumulado es E. El calculador, en la línea 26 lo que hace es restar de ese total, la última carta sacada (C), creando una nueva variable $F = E - C$, que será el valor que habrá de superar. Este paso es anulado por dos causas:

a) Que sea F menor que 5 (puedes cambiar el 5 por otro valor que tú creas más conveniente). Esto ocurre, por ejemplo, cuando la primera carta que te sale es un 6 o un 7 y decides (muy sabiamente) plantarte, o cuando teniendo un 2 te sale un 5. En tales casos, el calculador se obliga a sacar más de 5.

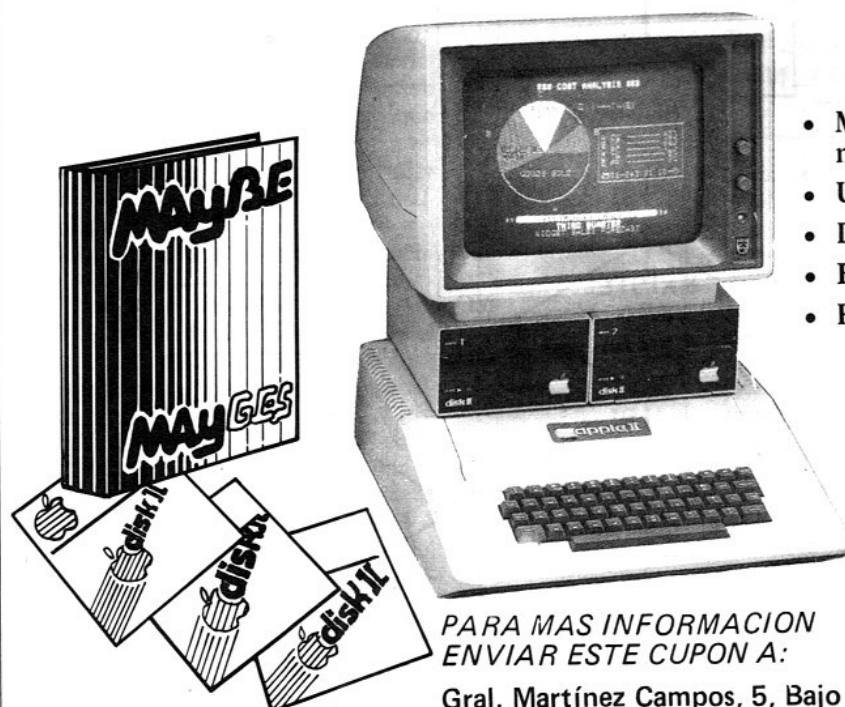
b) Esta causa constituye la 2ª excepción a la regla de la carta oculta, esto es, cuando te plantas con siete y media. Cuando esto ocurre, el calculador, en la línea 27 se da por enterado y obra en consecuencia, haciendo que el mínimo F a superar sea $F = 7$ (habrá de sacar más de siete y no pasarse, luego sacar siete y media).

En el caso de que sea la máquina quien juega primero, la línea 38 (en este caso, el bloque empieza en la 37) contiene un método para generar la variable F y que nos dará un número com-



MAYGES + apple II =

CONTABILIDAD RESUELTA
SEGUN EL PLAN
CONTABLE NACIONAL.



- Muchísimos usuarios en España lo testimonian.
- Utilice el programa Mayges.
- Desarrolle con Apple II su contabilidad.
- Entrega inmediata.
- Precio: 57.500 pts.

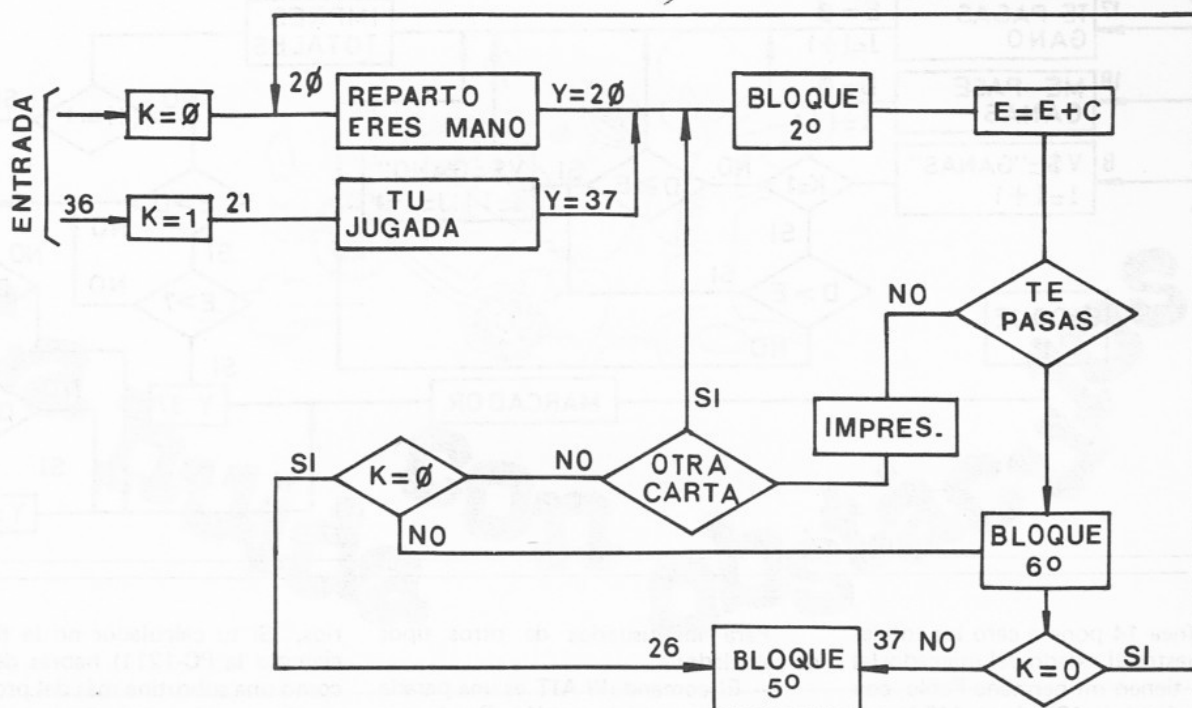
PARA MAS INFORMACION
ENVIAR ESTE CUPON A:

Gral. Martínez Campos, 5, Bajo izda.
MADRID-10 - Tfnos.: 445 84 38 - 446 60 18

Brusi, 102, Entresuelo 3º
BARCELONA - 6 - Tfnos.: (93) 201 21 03

NOMBRE
DIRECCION
TFNO.
CIUDAD

BLOQUE 4º



prendido entre 3 y 5. De esta forma queda más aleatorio el total acumulado por el calculador. Si no te gusta el que hay, puedes cambiar este método por otro más acorde con tus preferencias.

La salida de este 5º bloque se hace por dos vías, según quién sea mano:

a) Si quien reparte es la máquina, ($K = 0$), la línea 33 trasfiere el control del programa al bloque 6º, donde se decide el ganador.

b) Si la máquina es mano ($K = 1$), las líneas 35 y 36 se encargan de esconder su última carta (salvo en el caso de que tenga siete y media, ya que entonces

lo que hace es esconder su última carta pero avisarle de lo que tiene mediante "TOT: 7.5") y, tras la frase "TE TOCA JUGAR", transferir el control a la línea 21 (al bloque 4º).

BLOQUE 6º.- Este bloque es el encargado de decidir quien ha ganado esta mano y quién va a repartir en la siguiente (recuérdese que el que tiene la baraja gana la mano, con, tan solo, empatar la puntuación del contrario), indicando además los puntos de cada uno y el estado del marcador. Está formado por las líneas 8 a 18 salvo las 15 y 16 que forman el bloque 1º.

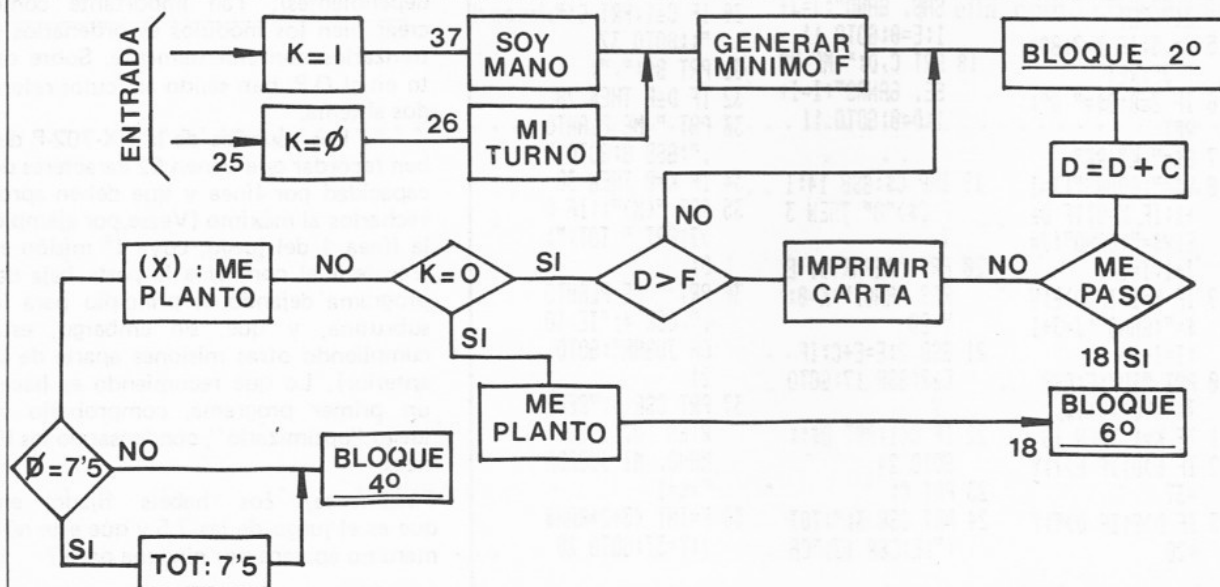
Se entra por dos caminos, según motivo:

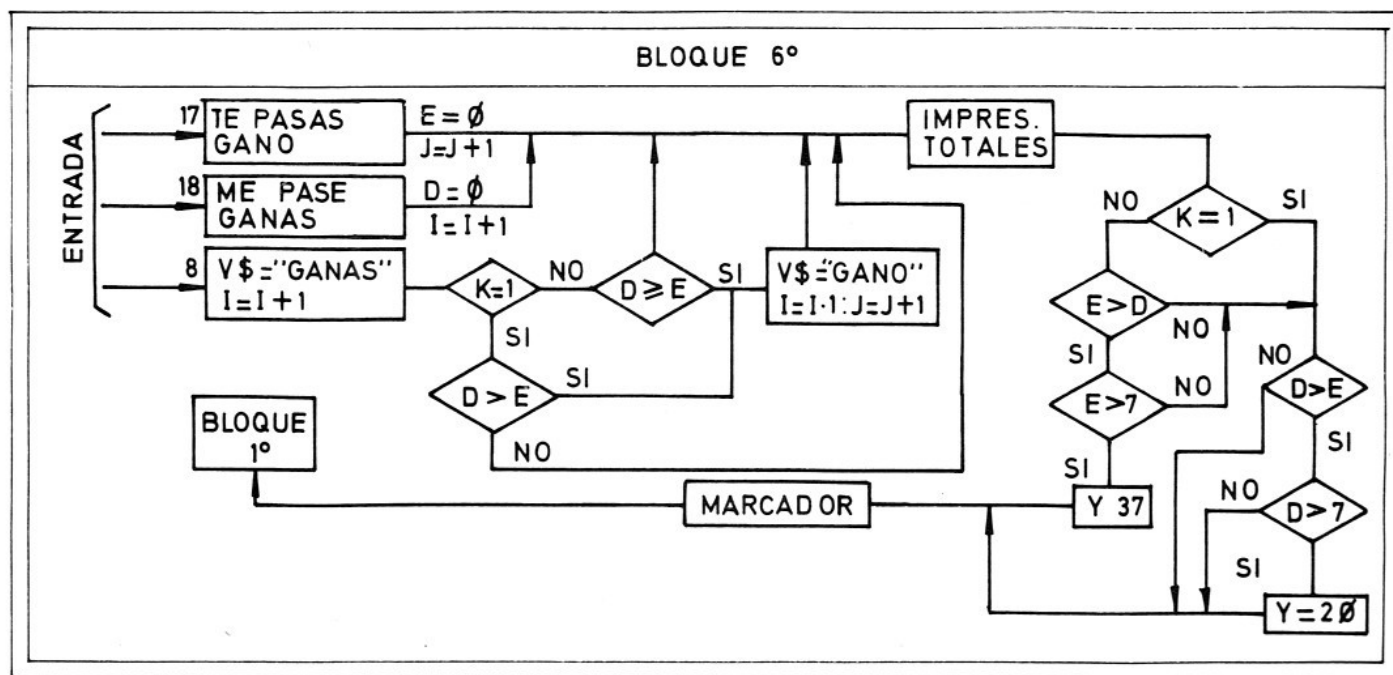
a) Si uno de los dos jugadores se pasa, se entra al bloque por las líneas 17 ó 18 si es la máquina la que lo ha hecho).

b) Si no se ha pasado nadie, la línea de entrada al bloque es la 8, decidiéndose en ésta y las dos siguientes quien es el ganador de la mano. La línea 10 nos muestra los totales obtenidos, diciendo quien gana.

Ambas entradas desembocan en la línea 11. Esta y las dos siguientes tienen como misión adjudicar la baraja en la siguiente mano.

BLOQUE 5º





La línea 14 pone a cero los totales y nos muestra el estado del marcador (el record lo tienen mi hermano Pablo, con un tanteo final de 127 el por 113 la máquina, transfiriendo el control al bloque 1° que comprueba el estado de la baraja y la repone en caso necesario. Desde aquí, el programa saltará al punto de retorno, el cual, a su vez, lo enviará al lugar indicado por las líneas 11, 12 y 13.

Para los usuarios de otros tipos de calculador:

— El comando **W AIT** es una parada de programa para impresión. Puede que ellos tengan un **PAUSE** u otro equivalente.

— El **CSR**, por su parte, es un control del cursor totalmente análogo al **TA B** de la mayoría de las máquinas.

— **RAN#** es una rutina interna para generación de números pseudoaleato-

rios. Si tu calculador no la tiene (por ejemplo la PC-1211) habrás de ponerla como una subrutina más del programa.

A nivel de reglas del juego, hay una que no aparece en el programa: cuando la primera carta que sale es un cuatro, el que juega tiene derecho a devolverla y pedir otra en su lugar. Cuando creé el programa, me pareció que esta norma restaba emoción al juego (pido...no pido...) y decidí no incluirla. Vosotros podéis hacerlo si queréis.

En el punto de economías de programación (recordad que inicialmente era 400 pasos más largo) puedo apuntar varias notas que, a lo mejor os sirven de ayuda en otros programas de vuestra CASIO:

— Este programa está hecho en lo que podemos llamar estructura modular. Esta técnica consiste en dividir el programa en bloques, cada uno con una misión específica, y que se pueden crear y comprobar por separado (rutinas independientes). Tan importante como crear bien los módulos es ordenarlos y trenzarlos convenientemente. Sobre esto en el O.P. han salido artículos referidos al tema.

— Los usuarios de la FX-702-P deben recordar que tienen 62 caracteres de capacidad por línea y que deben aprovecharlos al máximo (Véase por ejemplo la línea 1 del juego, cuya 1ª misión es transferir el control a la parte baja del programa dejando el principio para la subrutina, y que, sin embargo, está cumpliendo otras misiones aparte de la anterior). Lo que recomiendo es hacer un primer programa, comprobarlo y, luego "optimizarlo", condensando las líneas.

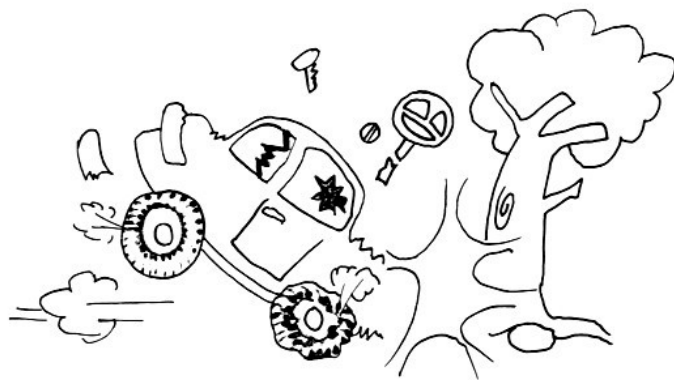
Finalmente, ¿os habeis fijado en que es el juego de las 7.5 y que este número no aparece por ninguna parte?

Joaquín E. Izquierdo Sobrino.

```

P2: 950 STEPS
1 VAC :WAIT 60:PR
T " SIETE Y M
EDIA", "QUIERES
REPARTIR":GOTO
19
2 Z=INT (RAN#*10)
:C=Z+1:IF Z=10
THEN 2
3 IF A(Z)=Ø THEN
2
4 A(Z)=A(Z)-1:IF
Z<7:RET
5 C=.5:IF Z=7:B$=
" J":RET
6 IF Z=8:B$=" Q":
RET
7 B$=" K":RET
8 V$="GANAS":I=I
+1:IF K=Ø:IF D≥
E:V$="GANO":J=
J+1:I=I-1
9 IF K=1:IF D>E:V
$="GANO":J=J+1
:I=I-1
10 PRT "TU":E:CSR
7:"YØ":D:V$
11 IF K=1 THEN 13
12 IF E>D:IF E>7:Y
=37
13 IF D>E:IF D>7:Y
=20
14 D=Ø:E=Ø:PRT "VA
ROS: TU":I:" Y
Ø":J
15 M=Ø:FOR L=Ø TO
9:M=M+A(L):NEXT
L:IF M≥15:RET
16 PRT "==ESTOY BA
RAJANDO==":FOR
L=Ø TO 9:A(L)=4
:NEXT L:RET
17 PRT C:E:"-TE PA
SAS. GANO":J=J+
1:E=Ø:GOTO 11
18 PRT C:D:"-ME PA
SE. GANAS":I=I+
1:D=Ø:GOTO 11
19 INP C$:GSB 14:I
F C$="Ø" THEN 3
7
20 PRT "REPARTO. E
RES MANO.":K=Ø:
Y=20
21 GSB 2:E=E+C:IF
E≥8:GSB 17:GOTO
Y
22 IF C<1:PRT B$:I
GOTO 24
23 PRT C:
24 PRT CSR 3:"TOT
I":E:CSR 13:"CA
RTA":INP A$:IF
A$="Ø" THEN 21
25 IF K=1:GSB 8:GO
TO Y
26 PRT CSR 6:"MI T
URNO":F=E-C:IF
5>F:F=5
27 IF E>7:F=7
28 GSB 2:D=D+C:IF
D≥8:GSB 18:GOTO
Y
29 IF D>F THEN 34
30 IF C≥1:PRT C:
":GOTO 32
31 PRT B$:",
32 IF D≠F THEN 28
33 PRT " ME PLANTO
.":GSB 8:GOTO Y
34 IF K=Ø THEN 30
35 PRT "(X)":IF D
>7:PRT " TOT":
D:
36 PRT " ME PLANTO
.":CSR 4:"TE TO
CA JUGAR":GOTO
21
37 PRT CSR 4:"REPA
RTES TU.":SOY
MANO. MI JUGADA
":K=1
38 F=INT (3+2*RAN#
):Y=37:GOTO 28

```



La huida con obstáculos

¿Es EL BASIC intérprete muy lento para realizar un juego de video?. Este programa demuestra que no. Sólo es necesario evitar complicaciones inútiles para avanzar a toda velocidad, por un recinto lleno de obstáculos. Buena Suerte.

En la pantalla del TRS 80 se distingue un punto que parpadea. Es la posición de salida del vehículo. Antes de arrancar, respire hondo y concéntrese, ya que no hay freno. Para detenerse no hay más que una solución: chocar contra algo. Pero el juego termina. Al cabo de cuánto tiempo? El resultado que aparece se lo comunica.

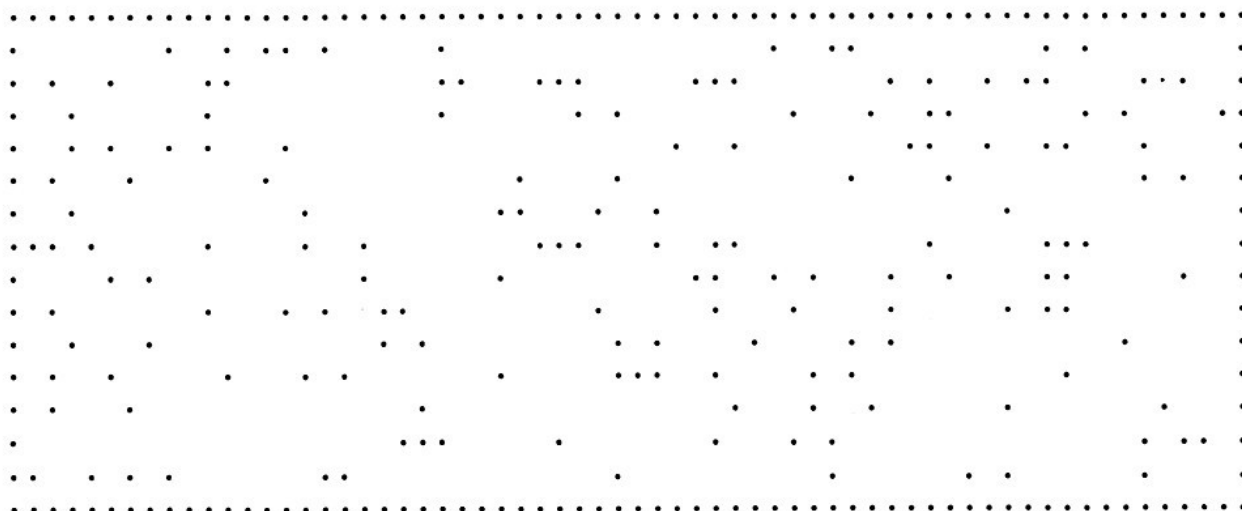
Debe evolucionar evitando los obstáculos diseminados por el recinto. Están representados por puntos luminosos situados al azar. Pero cuidado, no puede tomarse dos veces el mismo camino. Elija con cuidado la trayectoria, siempre que tenga tiempo de hacerlo.

El programa empieza por preguntar el número de obstáculos y la velocidad. Si se empeña, contra toda lógica, en elegir la máxima velocidad (100 kms/h.), es usted muy dueño.

Con una velocidad baja, el juego está al alcance de los niños. Hacen progresos rápidamente.

Para desplazarse se utilizan las teclas ↑ ↓ → ←; cuando se pulsa una de ellas, se continúa en esa dirección hasta que se cambie de dirección o el vehículo colisione.

Se considera colisión, la efectuada contra los límites del recinto, los obstáculos y la trayectoria pasada (tanto volviendo sobre ella, como cruzándola).




```

10 REM      * SORTEO DE OBSTACULOS *
20 REM  AUTOR: JESUS GARCIA
30 REM  COPYRIGHT EL ORDENADOR PERSONAL
40 REM  *****
100 GOSUB 1000      :REM REGLAS DE JUEGO
110 GOSUB 1200      :REM CONDICIONES INICIALES
120 GOSUB 1300      :REM PREPARACION DEL TERRENO
140 GOSUB 1500      :REM ESPERA DE UN CARACTER
150 GOSUB 1700      :REM DESPLAZAMIENTOS
160 GOSUB 1900      :REM RESULTADO
170 PRINT
180 PRINT"QUIERE JUGAR OTRA PARTIDA EN LAS MISMAS CONDICIONES";
190 GOSUB 9000      : REM TEST SOBRE LA RESPUESTA
200 IF RS="S" THEN 120 ELSE 110
210 END

1000 REM *****
1010 REM      REGLAS DE JUEGO
1020 CLS
1030 PRINT TAB(10)"**** SORTEO DE OBSTACULOS ****"
1040 PRINT TAB(10)"-----"
1050 PRINT:PRINT"PULSAR SOBRE  ";CHR$(93);
1060 PRINT"PARA DESPLAZARSE HACIA LA IZQUIERDA"
1070 PRINT:PRINT" ";CHR$(94);
1080 PRINT"----- LA DERECHA"
1090 PRINT:PRINT" ";CHR$(91);
1100 PRINT"----- ARRIBA"
1110 PRINT:PRINT" ";CHR$(92);
1120 PRINT"----- ABAJO"
1130 PRINT:PRINT
1140 PRINT"TRATAR DE EVITAR LOS OBSTACULOS! BUENA SUERTE"
1150 RETURN
1200 REM -----
1210 REM      CONDICIONES INICIALES
1220 PRINT:PRINT
1230 INPUT"VELOCIDAD DE 1 A 100 KM/H";V
1240 V=101-V
1250 IF V<1 OR V>100 THEN 1230
1260 INPUT"NUMERO DE OBSTACULOS DE 100 A 500";OB
1270 IF OB<100 OR OB>500 THEN 1260
1280 RETURN
1300 REM -----
1310 REM      PREPARACION DEL TERRENO
1320 RANDOM
1330 CLS: C=0
1340 FOR X=0 TO 127 :REM TRAZA DE BORDES DEL TERRENO
1350   SET(X,0)
1360   SET(X,47)
1370 NEXT X
1380 FOR Y=0 TO 47
1390   SET(0,Y)
1400   SET(127,Y)
1410 NEXT Y
1420 FOR I=0 TO OB :REM SITUA LOS OBSTACULOS
1430   SET(RND(126),RND(46))
1440 NEXT I : REM GENERACION DEL VEHICULO
1450 X=RND(126):Y=RND(46)
1460 IF POINT(X,Y)=-1 THEN 1450
1470 RETURN
1500 REM -----
1510 REM      ESPERA DE UN CARACTER
1520 Z=0
1530 RS=INKEY$
1540 Z=Z+1
1550 IF Z=15 THEN GOSUB 1600
1560 IF RS="" THEN 1530
1570 RETURN
1600 REM -----
1610 REM      PARPADEO DEL PUNTO
1620 Z=0
1630 IF POINT(X,Y)=-1 THEN RESET(X,Y) ELSE SET(X,Y)
1640 RETURN
1700 REM -----
1710 REM      DESPLAZAMIENTOS
1720 B=ASC(RS)
1730 IF B=8 THEN M=-1 : N=0
1740 IF B=9 THEN M=1 : N=0
1750 IF B=91 THEN M=0 : N=-1
1760 IF B=10 THEN M=0 : N=1 ELSE 1820
1770 X=X+M : Y=Y+N
1780 IF POINT(X,Y)=-1 THEN 1840
1790 C=C+1
1800 SET(X,Y)
1810 FOR T=1 TO V : NEXT T
1820 RS=INKEY$
1830 IF RS="" THEN 1770 ELSE 1720
1840 RETURN
1900 REM -----
1910 REM      RESULTADO
1920 FOR T=1 TO 1000 : NEXT T
1930 CLS
1940 PRINT "VUESTRA PUNTUACION ES DE";C;" PUNTOS"
1950 PRINT
1960 IF C<=ZZ THEN PRINT""A MEJOR PUNTUACION ES LA DE ";N$
1961 PRINT" CON ";ZZ;" PUNTOS": GOTO 2000
1970 PRINT "BRAVO;ES USTED EL MEJOR"
1980 PRINT:INPUT"CUAL ES SU APELLIDO ";N$
1990 ZZ=C
2000 RETURN
2010 REM -----
2020 REM      TEST SOBRE LA RESPUESTA
2030 RS=INKEY$
2040 IF RS="" THEN 9020
2050 IF RS<>"S" AND RS<>"N" THEN PRINT" SI O NO ";:GOTO 9020
2060 RETURN

```

Los invasores han vuelto, yo los he encontrado

¿Quién no conoce a los invasores? Hoy tiene Vd. que vérselas con invasores nada "corrientes": hordas de extraterrestres, a la cabeza de los cuales reina una cazador invencible. Este último tiene, en efecto, la posibilidad de "tragarse" los tiros que le son disparados. Su propósito: reducirle a la nada. Pero el Vic 20 viene en su ayuda y quizás tenga éxito eliminándolos antes de que ellos le destruyan.

Para intentar salvar la tierra (y a Vd. mismo, al tiempo), dispone de un cañón móvil cuyo manejo es muy simple:

- la tecla CRSR para maniobrar hacia la izquierda,
- la tecla CRSR para maniobrar hacia la derecha,

- la barra de espaciamento para disparar.

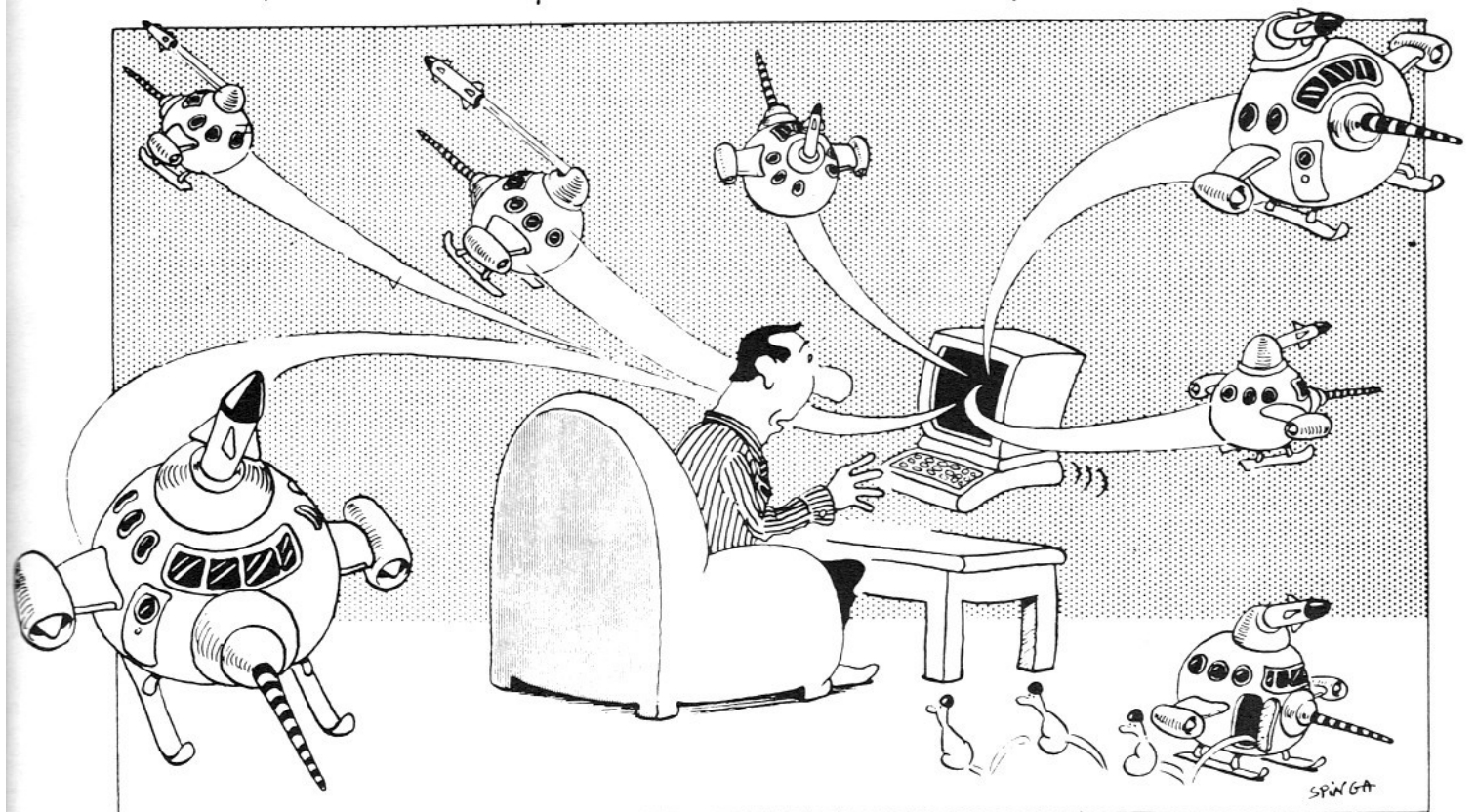
¿Esto es todo, me dirá Vd.? si, bueno casi, pues le falta introducir el programa. Luego, le toca jugar.

Si se quiere jugar evitando releer las explicaciones, es preciso pulsar RUN50.

Principales variables

P: posición del cañón.
PC: posición del cazador.
T1, T2: indicadores de tiro.
NT: número de tablas jugadas.
IV: incremento de la posición del cañón.
IC: incremento de la posición del cazador.
TV: posición de tiro del cañón.
LT: posición de tiro del cazador.
S: puntuación.

El defecto principal de este programa es su relativa lentitud debida al lenguaje utilizado, el Basic. Además, como ocupa la casi totalidad de la me-



READY.

Francisco Dervaux

CASIO

LE OFRECE LAS MEJORES
SOLUCIONES A SUS
MÚLTIPLES
PROBLEMAS
CON SU

FP-200

ORDENADOR PROFESIONAL PORTÁTIL



➡ ...y a un precio increíble: 79.500 pts.

AGENDA/DIETARIO/CONTABILIDAD
— Permite archivar unos 200 clientes.
— Realizar unos 1000 asientos contables.
— Llevar un diario completo.

MATEMÁTICAS II
— Interpolación lineal y parabólica. — Resolución de ecuaciones. — Ajustes por mínimos cuadrados. — Cambios de base. — Polinomios.

MATEMÁTICAS I
— Derivadas 1ª y 2ª.
— Integrales
— Máximos y mínimos
— Resolución $F(x) = 0$

BASE DE DATOS
— Le permite crear cualquier tipo de fichero, definiendo los campos que Vd. desee.

FACTURACIÓN Y ARCHIVO DE CLIENTES
— Permite archivar unos 300 clientes con todos sus datos.
— Almacenar de 25 a 50 facturas, junto con sus albaranes, al introducir los datos del pedido y gestionar el archivo de clientes.
— También permite efectuar las facturas y albaranes individualmente.

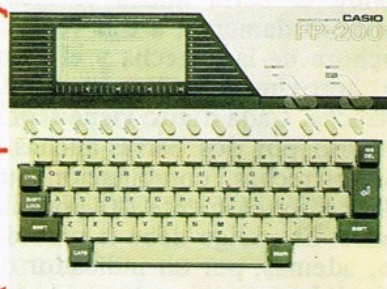
TRATAMIENTO DE TEXTOS CON GESTIÓN DE FICHERO DE CLIENTES
— Permite crear, aumentar, corregir, editar, archivar, cargar e imprimir un texto y gestiona el archivo de clientes.

JUEGOS I
— La gallina saltarina
— El gusano loco
— Batalla de tanques

JUEGOS II
— Slalom
— Batalla naval
— Muertos y heridos

ARQUITECTURA
— Pilares metálicos
— Viga biapoyada
— Cerchas

LENGUAJES
— Dissassembler
— Volcado hexadecimal



ELCO1

Electrónica de consumo-1, S.A. (Grupo OTESA)
c/ Virgen de Lourdes, 40 post. 28027-MADRID
Telfnos. 405 02 00 y 405 02 61

Nombre
Dirección
Población
Profesión
Desearía recibir información más amplia sobre el FP-200.

Códigos de barras.

Con la impresora C. ITOH 8510.



En el presente artículo se describe lo que es un código de barras, tomando como ejemplo el denominado "Industrial 2 de 5" y la forma de llevarlo a cabo mediante una impresora de agujas (la C. Itoh 8510) y un sencillo programa Basic.

¿Qué es un código de barras?

Por código de barras se entiende normalmente un sistema de codificación, que permite introducir datos directamente al ordenador mediante un lector óptico (luz, laser o infrarrojos). En esta codificación, cada dígito o carácter viene representado por una sucesión de barras negras, variables en anchura y/o en número y separadas por espacios blancos, que el ordenador "lee" interpretándolo según unas reglas determinadas. Actualmente se utiliza en multitud de aplicaciones. Por ejemplo, en los grandes almacenes, cada punto de venta dispone de un lector óptico y cada artículo lleva grabado previamente el código que lo identifica. A la hora de la venta, el operador desliza el lector sobre la superficie del código de barras y el ordenador se entera inmediatamente de qué artículo está siendo vendido, con lo cual imprime la factura, actualiza existencias, etc. de forma automática y sin posibilidad de error.

Código industrial 2 de 5.

Este código permite representar cada uno de los 10 dígitos decimales en sucesión ilimitada. Está formado por barras de dos anchuras distintas, la ancha y la estrecha, separadas por blancos, donde la barra ancha es igual, aproximadamente, a tres veces la anchura de la estrecha y el blanco, aproximadamente, igual a la estrecha. Cada dígito viene representado por dos barras anchas y tres estrechas, combinación que admite exactamente 10 posibilidades. El código debe ir precedido, además, por un indicador de principio (start) y un indicador de final (end) para la sincronización y para permitir la lectura en ambos sentidos. En el programa basic puede verse cómo se codifica cada dígito, así como los dos indicadores.

Programa Basic:

El programa Basic presentado en la figura, permite construir

mediante códigos 2 de 5 mediante la impresora C. Itoh 8510. Está dividido en tres partes: inicialización de variables, input por pantalla, y salida por impresora.

Las líneas 10230-10320 constituyen la definición del código. La variable UNOS\$ representa una barra ancha más un espacio blanco y la variable CEROS\$ una barra estrecha más un blanco. La variable dimensionada CODE\$() contiene las codificaciones de cada dígito, de forma que CODE\$(0) contiene el código para el 0, CODE\$(1) para el 1, etc.

En la línea 10380 se entra por pantalla el número que se desea codificar y se deposita en la variable string CODE\$.

Las líneas de la 10450 a la 10600, que acaban con un GOTO, podrían acabar con un RETURN y constituir de esta forma una subrutina de utilidad general. Las variables de entrada que utilizaría serían, la CODE\$(string a codificar) y la P025 (columna o posición a imprimir el código).

Los pasos que efectúa esta subrutina son los siguientes:

La instrucción 10450 pone a cero el espaciado interlíneas (line feed de 16/144 pulgadas).

La 10460 coloca a la impresora en unidireccional. Esto es conveniente para obtener mayor homogeneidad de impresión, aunque no resulta imprescindible.

La línea 10470 coloca a la impresora en "boldface", letra negra con la que se consigue mayor densidad de impresión. Tampoco es imprescindible si la cinta está en buen estado.

El bucle FOR-NEXT de la línea 10480 es para repetir 4 veces la misma línea, consiguiéndose de esta forma un código suficientemente ancho.

La línea 10490 es el comando "start graphics mode". En graphics mode esta impresora inhibe su generador interno de caracteres y pasa a bit image. En bit image cada bit de un byte representa a una de las ocho agujas de impresión. Si el bit está a uno sig-

Programa.

```

10000 /
10010 /
10020 / FICHERO BAR2/5.BAS
10030 / PROGRAMA CODIGO DE BARRAS
10040 / CODIGO INDUSTRIAL 2/5
10050 /
10060 / PROGRAMA HECHO POR F.H.C.
10070 / DISTRIBUIDORA DE SISTEMAS EL.
10080 /
10090 / ORDENADOR : SUPERBRAIN
10100 / CP/M 2.2
10110 / MBASIC 5.21
10120 /
10130 / IMPRESORA : C.I.TOH 8510
10140 /
10150 /
10160 /
10170 / INICIALIZACION DE VARIABLES:
10180 /
10190 UNO$=STRING$(3,255)+STRING$(2,0)
10200 CERO$=STRING$(1,255)+STRING$(2,0)
10210 DIM CODE$(9)
10220 WIDTH LPRINT 255
10230 CODE$(0)=CERO$+CERO$+UNO$+UNO$+CERO$
10240 CODE$(1)=UNO$+CERO$+CERO$+CERO$+UNO$
10250 CODE$(2)=CERO$+UNO$+CERO$+CERO$+UNO$
10260 CODE$(3)=UNO$+UNO$+CERO$+CERO$+CERO$
10270 CODE$(4)=CERO$+CERO$+UNO$+CERO$+UNO$
10280 CODE$(5)=UNO$+CERO$+UNO$+CERO$+CERO$
10290 CODE$(6)=CERO$+UNO$+UNO$+CERO$+CERO$
10300 CODE$(7)=CERO$+CERO$+CERO$+UNO$+UNO$
10310 CODE$(8)=UNO$+CERO$+CERO$+UNO$+CERO$
10320 CODE$(9)=CERO$+UNO$+CERO$+UNO$+CERO$
10330 /
10340 /
10350 /
10360 / INPUT POR PANTALLA:
10370 /
10380 LINE INPUT "ENTRAR NUMERO >";CODE$
10390 IF CODE$="" THEN END
10400 /
10410 /
10420 /
10430 / SALIDA IMPRESORA:
10440 /
10450 LPRINT CHR$(27) "T16";
10460 LPRINT CHR$(27) ">";
10470 LPRINT CHR$(27) "!" CHR$(13);
10480 FOR IT25 = 1 TO 4
10490 LPRINT TAB(PO25) CHR$(27) "S";
10500 LPRINT RIGHT$(STR$(10000+LEN(CODE$)*19+26),4);
10510 LPRINT UNO$ UNO$ CERO$;
10520 FOR CN25=1 TO LEN(CODE$)
10530 LPRINT CODE$(VAL(MID$(CODE$,CN25,1)));
10540 NEXT CN25
10550 LPRINT UNO$ CERO$ UNO$
10560 NEXT IT25
10570 LPRINT CHR$(27) "T04" CHR$(27) "<";
10580 LPRINT CHR$(27) CHR$(34)
10590 LPRINT TAB(PO25) "*" CODE$ "*" CHR$(27) "A"
10600 LPRINT:GOTO 10380
10610 /
10620 /

```

Resultado.



nifica impacto de la aguja correspondiente; si está a cero significa no impacto. De esta forma se pueden confeccionar gráficos y caracteres especiales.

La instrucción 10500 le dice a la impresora cuántos bytes le mandaremos en bit image. Dado que para cada dígito se necesitan 19 bytes, en total necesitaremos (19 x longitud en caracteres de CODE\$) + los correspondientes a START y END, que son 26.

La instrucción 10510 envía un código START, el bucle FOR-NEXT de las líneas 10520-10540 envían sucesivamente la codificación de los dígitos contenidos en CODE\$, y la línea 10550 envía el indicador de final o END.

En la línea 10570, se define un nuevo espaciado interlíneas y se regresa a impresión bidireccional.

La 10580 termina la impresión en negrita para volver a densidad normal y la 10590 imprime, debajo de las barras y ligeramente separado, la representación en decimal del código en cuestión. A continuación, mediante la instrucción chr\$(27) "A", se regresa a un espaciado interlíneas normal (1/6 de pulgada). □

ORIC



LA DIFERENCIA ENTRE UN ORDENADOR Y UN JUGUETE PERSONAL

PRECIO UNIDAD CENTRAL
59.900 ptas.

Descubra con Oric todas las posibilidades que tiene un auténtico ordenador personal por capacidad, software y periféricos.

48 K. Más de treinta títulos de juegos.

Aplicaciones profesionales para el nuevo Microdisc.

Impresora Plotter con cuatro colores y, por supuesto, teclado profesional.

La única cosa de juguete en un Oric es el precio. Compruébelo.

Distribuidor exclusivo

TEXTRONIC-SPAN

Preciados, 39 - Madrid-13 - Teléfono 248 56 35

Gran premio de Penches



Una vez fijado el precio de la apuesta, los apostantes irán dando su nombre, así como el número de caballo sobre el cual quieren apostar, todo ello a petición del ordenador, pudiendo apostar por cualquier caballo tantas veces como se desee.

No podrán existir más de diez apostantes por caballo, en el momento de sobrepasar esta cifra, el ordenador nos dará el mensaje correspondiente, teniendo aún posibilidad de apostar por cualquiera de los otros caballos.

Una vez dado el mensaje de fin, no se podrán realizar más apuestas, en este momento

aparecerá el hipódromo en pantalla y los caballos se colocarán en sus posiciones de salida.

Pulsando la tecla "S" daremos comienzo al GRAN PREMIO DE PENCHES.

A la llegada y para deshacer cualquier posible duda sobre quién es el caballo ganador, el ordenador nos dirá quién es el caballo ganador y quiénes los ganadores, así como la cantidad del premio en pesetas.

La cantidad repartida en premios supone el 80% del montante total, puesto que el 20% es retenido por la organización.

Si no hubiera ninguna apuesta sobre el caballo campeón, la totalidad del montante apostado iría a parar a manos de la organización para mejoras en el hipódromo, así como si del resultado del reparto del 80% del montante total de apuestas, correspondiese una cantidad no entera en pts. a cada acertante, estas pts. irían también a parar a la organización. Por todo ello se les dará las gracias.

La organización se despide hasta el próximo concurso en el que esperan tengan ustedes más suerte.

Jesús García Cibrian.


```

10 REM ***** GRAN PREMIO DE PENCHES *****
20 REM AUTOR :JESUS GARCIA CIBRIAN
30 REM COPYRIGHT EL ORDENADOR PERSONAL Y EL AUTOR
40 DIM X(5),P$(5,10),R%(5)
50 CLS : P=0
60 PRINT TAB(18) "GRAN PREMIO DE PENCHES":PRINT:PRINT
70 PRINT TAB(19)"SE ABREN LAS APUESTAS":PRINT:PRINT
80 PRINT"PUEDEN USTEDES APOSTAR POR LOS 5 CABALLOS"
90 FOR I=1 TO 5 : X(I)=0 : R%(I)=0 : NEXT I
100 INPUT"CUANTO ES EL VALOR DE LA APUESTA";A : CLS
110 INPUT"NOMBRE DEL APOSTANTE(O FIN)";A$
120 IF A$="FIN" THEN 210
125 P=P+1
130 INPUT"SOBRE QUE CABALLO APUESTA";B
140 IF B<1 OR B>5 THEN 130
150 REM ***** GUARDAR NOMBRES DE APOSTANTES *****
160 R%(B)=R%(B)+1
170 IF R%(B)>10 THEN 176
175 P$(B,R%(B))=A$ : GOTO 110
176 R%(B)=R%(B)-1
180 PRINT"HAY YA 10 APOSTANTES PARA EL CABALLO";B
190 PRINT"NO PUEDE USTED APOSTAR POR ESE CABALLO"
200 P=P-1 : GOTO 110
210 CLS
220 REM ***** TRAZA DE LA PISTA *****
230 FOR X=0 TO 127
250     FOR Y=17 TO 42 STEP 5
260         SET(X,0+Y)
270     NEXT Y
280 NEXT X
290 REM ***** COLOCA LOS CABALLOS EN LA SALIDA *****
300 FOR Y=20 TO 40 STEP 5
310     SET(0,0+Y)
320 NEXT Y
330 PRINT@ 960 ,"PULSAR LA 'S' PARA DAR LA SALIDA"
340 A$=INKEY$ : IF A$(">"S") THEN 340
350 REM ***** MUEVE LOS CABALLOS *****
360 FOR J=1 TO 127
370     I=0
380     FOR Y=17 TO 37 STEP 5
390         I=I+1
400         RESET(X(I),Y)
410         X(I)=X(I)+INT(RND(3))
420         SET(X(I),Y)
430         IF X(I)>=127 THEN Y=37 : J=127
440     NEXT Y
450 NEXT J
460 REM ***** CONOCIMIENTO DEL GANADOR *****
470 PRINT"EL CABALLO GANADOR ES EL";I
480 D=R%(I)
490 IF D=0 THEN V=A*P : GOTO 600
500 FOR H=1 TO D
510     PRINT P$(I,H);" ";
520 NEXT H
530 PRINT"GANADOR";:IF D>1 THEN PRINT"ES";
540 PRINT" DEL GRAN PREMIO DE PENCHES"
550 N=((A*P)*2)/10
560 R=((A*P)-N)/D : Z=INT(R) : V=(R-Z)*D
570 PRINT"PREMIO DE";Z;"POR ACERTANTE"
580 PRINT"CANTIDAD RETENIDA POR LA ORGANIZACION";N;"PTS"
600 IF V(>"0") THEN PRINT"GRACIAS POR LAS";V;"PTS"
610 PRINT"HASTA EL PROXIMO CONCURSO"
630 INPUT"QUIERE JUGAR OTRA PARTIDA (SI O NO)";A$
640 IF A$(">"SI") AND A$(">"NO") THEN 630
650 IF A$="SI" THEN 50
660 END

```


juego

Programas de 1 K para el ZX-81



DUELO

En esta ocasión Ud. se enfrenta en duelo con su ZX81. En la pantalla aparecerán dos números elegidos al azar. El de la izquierda corresponde al ZX81, el de la derecha es el suyo.

Si su número es mayor o igual al del ZX81, ataque con "5".

Si su número es menor, defiéndase con "8". Pero todavía hay otra posibilidad: si aparece la palabra "FUEGO", dispare sin pensarlo con "O".

No sólo debe no equivocarse en su decisión. Debe además ser más rápido que el ZX81... lo que no es fácil!

```

1000 LET H=VAL "3"
1010 LET O=CODE "O"
1020 LET D=CODE "8"
1030 LET U=D/D
1040 LET Z=D-D
1050 LET V=Z
1060 PRINT TAB D;Z;TAB D+D;U;AT
1070 LET E=Z
1080 LET S=U
1090 IF Z>D*D OR U>D*D THEN ST
1100 IF RAND<.2 THEN LET E=U
1110 LET A$=""
1120 LET A$(U)=STR$ INT (RAND*D)
1130 LET A$(H)=STR$ INT (RAND*D)
1140 LET T=INT (RAND*VAL "8")+U
1150 FOR G=0 TO T
1160 PRINT AT D,G;" ";
1170 LET S=D+D
1180 NEXT G
1190 IF E=U THEN LET A$="FUEGO"
1200 PRINT A$
1210 FOR A=U TO T+H
1220 LET J$=INKEY$
1230 IF J$<>" " THEN GOTO CODE "5"
1240 NEXT A
1250 LET Z=Z+VAL "5"
1260 PRINT AT 0,0;CODE "5";"-----"
1270 PRINT AT 0,0;"XX"
1280 FOR I=U TO H*0
1290 NEXT I
1300 CLS
1310 GOTO D+D
1320 IF (J$="5" AND A$(U)<A$(H))
1330 OR (J$="8" AND A$(U)>A$(H)) OR
1340 (J$="O" AND E=U) THEN GOTO CODE
1350 GOTO CODE "5"
1360 LET V=V+VAL "5"
1370 LET S=D
1380 GOTO CODE "5"

```

SALTO DE LA RANA

Ud. debe invertir la posición de las piezas, pasando todas las "X" al lado derecho y todas las "O" al lado izquierdo, en el menor número de jugadas posible.

Introduzca su jugada como un número de dos dígitos (origen/destino) y N/L. El ZX81 le mostrará inmediatamente la nueva situación, informándole además de su última jugada y el total de movimientos realizados.

SALTO DE LA RANA
POR ANTONIO MENENDEZ

```

1000 LET H=0
1010 LET P=VAL "3"
1020 LET O=VAL "8"
1030 LET R=VAL "7"
1040 LET S=VAL "9"
1050 DO
1060 A(I)=I
1070 LET I=0 TO S
1080 NEXT I
1090 LET A(R)=A(P)+A(R)
1100 LET B=A(O)+A(O)+A(R)+A(O)
1110 LET C=A(P)+A(P)+A(S)
1120 LET D=A(P)+A(O)+A(R)
1130 LET E=A(O)+A(O)+A(P)
1140 LET F=A(O)+A(O)+A(P)
1150 LET G=A(O)+A(O)+A(S)
1160 LET H=A(O)+A(O)+A(S)
1170 PRINT AT 0,P;H;" ";A;" ";B;
1180 A(P)=A(R);A(O)=A(O)+A(R)
1190 A(R)=A(O);A(P)=A(P)+A(R)
1200 A(O)=A(S);A(S)=A(S)+A(O)
1210 INPUT K
1220 INPUT J
1230 FOR I=0 TO S
1240 IF A(I)<>J THEN NEXT I
1250 IF A(L)<>J THEN NEXT L
1260 LET A(L)=J
1270 LET A(I)=K
1280 GOTO CODE "PI"

```

CUADRADO MAGICO 3X3

He aquí la versión computizada de un antiguo pasatiempo matemático de origen chino...

Ud. tiene un cuadrado de 3X3. En la posición de partida, contiene los números del 1 al 9 en orden ascendente. Ud. debe cambiar los números de posición en el interior del cuadrado para conseguir que la suma de los números de cada columna sea la misma..., e igual a la suma de cada línea..., e igual a la suma de cada diagonal.

No, no es imposible, pero entre las miles de combinaciones posibles, sólo existe **una correcta!**

Introduzca el primer número que quiere cambiar de posición, y N/L.

El 2º número a cambiar, y N/L. El ZX81 le mostrará la nueva situación del cuadrado, así como la suma de las columnas, diagonales y líneas para hacerle algo

más fácil su trabajo... Recuerde que para que los cálculos sean más rápidos, puede poner el ordenador en modo FAST antes de empezar el juego.

CUADRADO MAGICO 3X3

```

2000 LET O=PI/PI
2010 LET P=VAL "3"
2020 LET Q=VAL "8"
2030 LET R=VAL "7"
2040 LET S=VAL "9"
2050 DO
2060 A(I)=I
2070 LET I=0 TO S
2080 NEXT I
2090 LET A(R)=A(P)+A(R)
2100 LET B=A(O)+A(O)+A(R)+A(O)
2110 LET C=A(P)+A(P)+A(S)
2120 LET D=A(P)+A(O)+A(R)
2130 LET E=A(O)+A(O)+A(P)
2140 LET F=A(O)+A(O)+A(P)
2150 LET G=A(O)+A(O)+A(S)
2160 LET H=A(O)+A(O)+A(S)
2170 PRINT AT 0,P;H;" ";A;" ";B;
2180 A(P)=A(R);A(O)=A(O)+A(R)
2190 A(R)=A(O);A(P)=A(P)+A(R)
2200 A(O)=A(S);A(S)=A(S)+A(O)
2210 INPUT K
2220 INPUT J
2230 FOR I=0 TO S
2240 IF A(I)<>J THEN NEXT I
2250 IF A(L)<>J THEN NEXT L
2260 LET A(L)=J
2270 LET A(I)=K
2280 GOTO CODE "PI"

```

Los dos programas de 1 K. para el SINCLAIR ZX81 publicados en el nº 16 de Junio-1983 son de ANTONIO MENENDEZ, junto con estos forman un

grupo de interés para el estudio de cómo utilizar las características del BASIC Sinclair, para aprovechar el máximo de la memoria.

por ANTONIO MENENDEZ

PRESENTAMOS...

EL ORDENADOR MAS AVANZADO DEL MUNDO

SPECTRAVIDEO™

SV-318

MKII

SV-328



MSX

**MICROSOFT BASIC ★ FORTRAN 80 ★ Microsoft Logo ★ COLECO cartuchos
CP/M ★ Cobol 80 ★ MACROENSAMBLADOR Z80 ★ Ensamblador CP/M**

• MSX. Es el sistema escogido por 16 fabricantes japoneses de ordenadores (SONY, PANASONIC, PIONER, SANYO, SPECTRAVIDEO, etc.) siendo compatibles todos los programas creados por cada uno de los fabricantes.

CARACTERISTICAS GENERALES	SPECTRAVIDEO 328	SPECTRAVIDEO 318	APPLE II e	COMMODORE 64	ATARI 600 xL	BBC MODELO B	DRAGON 32	ORIC ATMOS
PROCESADOR	Z80 A	Z80 A	6502	6510	6502	6502	6809 E	6502
RELOJ EN MHz	4 MHz	4 MHz	1 MHz	1 MHz	1.8 MHz	2 MHz	1 MHz	1 MHz
MEMORIA STANDARD EN RAM	80 K	32 K	64 K	64 K	16 K	32 K	32 K	48 K
AMPLIABLE HASTA	144 K	144 K	—	—	64 K	—	64 K	64 K
MEMORIA STANDARD EN ROM	32 K	32 K	16 K	20 K	24 K	16 K	16 K	16 K
AMPLIABLE HASTA	96 K	96 K	—	—	—	—	—	—
MICROSOFT BASIC EXTENDIDO	SI	SI	—	—	—	NO	SI	SI
NUMERO DE TECLAS	87	51	62	66	57	73	53	57
TECLADO CON GRAFICOS PREDEFINIDOS	SI	SI	—	SI	SI	SI	NO	NO
TECLAS DE FUNCION PROGRAMABLES	10	10	—	8	4	10	NO	NO
MANDO JOYSTICK INCORPORADO	NO	SI	—	NO	NO	NO	NO	NO
COLORES	16	16	16	16	16	16	9	8
SPRITES	32	32	—	8	8	—	?	—
RESOLUCION (Puntos de pantalla)	256x192	256x192	280x192	320x200	320x192	256x640	256x192	240x200
TEXTO EN PANTALLA	40x24	40x24	40x24	40x25	40x24	40x32	32x16	40x28
CANALES DE SONIDO	3	3	1	3	4	1	3	3
OCTAVAS POR CANAL	8	8	4	9	?	3	5	8
A.D.S.R. ENVOLVENTE	SI	SI	NO	SI	NO	SI	NO	SI
CAPACIDAD UNIDAD DE DISCO	256 K	256 K	140 K	170 K	127 K	100 K	?	?
CENTRONICS	SI (opcional)	SI (opcional)	SI (opcional)	—	SI (opcional)	SI (opcional)	SI	SI
RS232	SI (opcional)	SI (opcional)	SI (opcional)	—	SI (opcional)	SI (opcional)	—	—
80 COLUMNAS	SI (opcional)	SI (opcional)	SI (opcional)	SI (opcional)	SI (opcional)	—	—	—
PRECIO P.V.P.	76.000	49.900	166.618	79.900	58.500	140.000	67.800	59.500

TM

— MSX y MICROSOFT son marcas registradas por MICROSOFT CORPORATION.
— CPM es una marca registrada por DIGITAL RESEARCH INC.
— COLECO VISION es una marca registrada por COLECO INDUSTRIES INC.

indescorp

HARDWARE
DISTRIBUIDOR OFICIAL DE SPECTRAVIDEO
INTERNATIONAL LTD
PASEO DE LA CASTELLANA, 179 - MADRID-16
TELEFONO (91) 279 31 05

DELEGACION EN CATALUÑA: ACE, S.A.
Tarragona, 100 - Barcelona-15.
Teléfono (93) 325 10 58

YA DISPONIBLE EN:



... Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

indescomp TE AYUDA A AUMENTAR LA POTENCIA DE TU

SV-318

SPECTRAVIDEO™

SV-328

PROGRAMAS



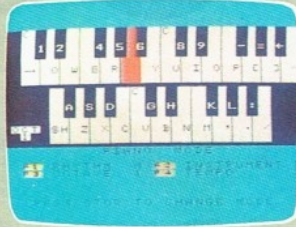
ENTRETENIMIENTO

- 220 C — SECTOR ALPHA
- 232 C — FRANTIC FREDDY
- 237 C — SUPER CROSS FIRE
- 291 C — FLIPPER SLIPPER
- 233 T — ARMoured ASSAULT
- 234 T — SPECTRA
- 219 T — NEW YORK BOMB BLITZ
- 302 T — 3 D TIC-TAC-TOE
- 309 T — SPECTRA BREAK
- 310 T — HORSE RACE
- 211 T — OLD MAC EL GRANJERO
- 216 T — TETRA HORROR
- 241 T — TELEBUNNY
- 242 T — TURBOAT
- 243 T — SASA EL ROBOT
- 292 T — NINJA EL GUERRERO
- 293 T — KUNG FU MASTER
- 245 T — LA PULGA (BUGABOO) 1
- 246 T — FRED
- 251 T — DONKEY KONG. COLECO
- 252 T — ZAXXON. COLECO
- 253 T — GORF. COLECO
- 254 T — CARNIVAL. COLECO
- 255 T — PITUFO. COLECO
- 256 T — TURBO. COLECO
- 261 T — OLIMPIC GAMES **MSX**
- 262 T — ANTARTIC ADVENTURE **MSX**
- 263 T — COMIC BAKERY **MSX**
- 264 T — MONKEY ACADEMY **MSX**
- 265 T — CIRCUS CHARLIE **MSX**
- 266 T — TIME PILOT **MSX**
- 267 T — ATHLETIC LAND **MSX**
- 268 T — SUPERCobra **MSX**



EDUCACION

- 209 T — WORD MASTER
- 213 T — STAR NUMBER
- 222 T — FIRST STEPS
- 304 T — ACUTYPE
- 303 T — NUMBER GAME
- 235 T — INTRODUC. AL BASIC
- 221 T — SWING MAN (AHORCADO)
- 331 T — CIUDADES DEL MUNDO
- 332 T — APRENDIENDO INGLES
- 333 T — TUTOR
- 334 T — SPECTRA ADIVINO
- 335 T — EL CUERPO HUMANO 1
- 336 T — EL LIBRO DE LOS RECORDS 1



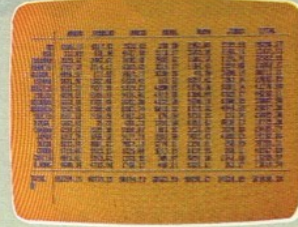
UTILIDADES

- 236 C — MUSIC MENTOR
- 294 C — JUST WRITE (PROC. TEXTOS)
- 266 T — SPRITE GENERATOR
- 520 D — FORTRAN 80
- 521 D — COBOL 80
- 522 D — MACRO ENSAMBLADOR Z80
- 523 D — ENSAMBLADOR CP/M
- 524 D — TURBO PASCAL
- 525 D — M. BASIC
- 526 D — COMPILADOR BASIC

T = Cassette

C = Cartucho

D = Diskette (CP/M)



PROFESIONALES

- 228 T — SPECTRA DIARY
- 229 T — SPECTRA HOME ECONOMIST
- 269 T — ADDRESS BOOK
- 275 T — SPECTRA FILE GABINET (FICHEROS)
- 227 T — SPECTRA CHECK BOOK
- 620 D — d. BASE II
- 622 D — REPORT STAR
- 623 D — SPELLSTAR
- 624 D — SUPERCALC
- 626 D — WORDSTAR
- 627 D — DATASTAR
- 628 D — CALCSTAR
- 633 D — MAIL MERGER
- 629 D — SUPERCALC 2
- 637 D — WORDSTAR & MALLMERGE
- 638 D — MULTIPLAN
- 639 D — CONTABILIDAD GENERAL (PLAN CONTABLE)
- 640 D — BASE DE DATOS
- 641 D — CONTROL STOKS
- 642 D — PROCESADOR TEXTOS
- 643 D — MAILING Y ETIQUETAS
- 644 D — ARQUITECTOS. PRESUPUESTOS
- 645 D — ARQUITECTOS. ESTRUCTURAS
- 646 D — VIDEOCLUBS

Todos los programas creados para los sistemas:

MSX

COLECO VISION**

CP/M* 2.2 y 3.0**

Son compatibles con los ordenadores **SPECTRAVIDEO**

* Con adaptador SV 606 MSX.

** Con adaptador SV 603 COLECO

*** Formato XEROX 820 II. 5 1/4

PERIFERICOS



SVI 605 A. EXPANDER CON DISCOS



PERIFERICOS



SVI 105 TABLETA GRAFICA



SVI 101 QUICKSHOT I y SVI 102 QUICKSHOT II

- SVI 904. DATACASSETTE
- SVI 803. 16K RAM
- SVI 805. INTERFACE RS 232
- SVI 806. CARTUCHO 80 COLUMNAS
- SVI 105. TABLETA GRAFICA

- SVI 602. MINIEXPANDER
- SVI 807. 64K RAM
- SVI 603. ADAPTADOR COLECO
- SVI. 909. IMPRESORA 80 COLUMNAS 130 c.p.s.
- SVI 606. ADAPTADOR MSX

- SVI 605 ó 605. SUPEREXPANDER CON DISCOS (256K ó 320 K FORMATEADOS)
- SVI 802. INTERFACE CENTRONICS
- SVI 101. QUICKSHOT I y SVI 102. QUICKSHOT II. MANDO PARA JUEGOS.

LIBROS

- GUIA de referencia Basic para el programador.
- SPECTRAVIDEO COMPUTING (inglés).
- M. BASIC. Manual de programación.
- SISTEMA OPERATIVO CP/M. Guía del usuario.
- SPECTRAVIDEO. Programación Avanzada 1
- INTRODUCCION A WORDSTAR.

Para mas información diríjanse a:
indescomp SPECTRAVIDEO
 P.º Castellana, 179 Telf. 279 31 05
 28016 MADRID

EXPOCOM



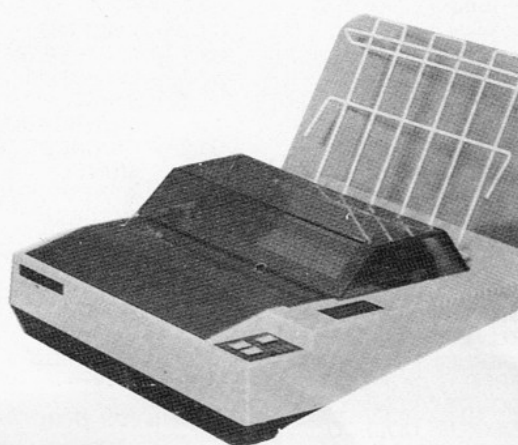
VILLARROEL, 68
TELS. (93) 3231933
2548813
BARCELONA-11

TOLEDO, 83
TEL. (91) 265 40 69
MADRID - 5

IMPRESORA CP-80

- * 80 COLUMNAS
- * 80 CARACTERES POR SEGUNDO
- * TRACCION / FRACCION
- * ENTRADA PARALELO CENTRONICS
- * OPCIONAL RS 232 C

- * GRAFICA
- * CARTUCHO TINTA
- * COMPATIBLE EPSON



 apple computers



APPLE COMPUTER INC.
DISTRIBUIDORES OFICIALES.
APPLE II/E
APPLE III
APPLE MACINTOSH
APPLE LISA 2
APPLE II/C

Software Apple

el ordenador en casa

No cabe duda de que entre los usos corrientes de un ordenador personal, el de los juegos es muy importante, no sólo desde un punto de vista recreativo sino también desde un punto de vista creativo, al ser éstos una fuente interminable de combinaciones, y por tanto, de estrategias.

Por desgracia, los juegos que dan pie a algoritmos interesantes desde un punto de vista técnico, no son abordables para el posesor, ya de un ordenador de posibilidades modestas, ya de una calculadora potente, como es el caso del SHARP PC-1211 (TRS 80 pocket), que se ven olvidados por las revistas especializadas como EL ORDENADOR PERSONAL: A ellos me dirijo proponiéndoles un programa-juego: el JUEGO DEL RADAR.

el juego del radar para SHARP PC-1211

El juego en sí.

En resumen, el juego consiste en lo siguiente:

El jugador dialoga con el ordenador que hace el papel, en el juego, de aparato de radar mermado en sus facultades (la pantalla del mismo estaría organizada como se muestra en la figura), y la misión del jugador es descubrir la posición de un avión que vuela en la zona barrida por dicho radar. El avión puede moverse como se muestra en la figura: de forma radial (1) o circular (2); en ningún caso se moverá con ambas componentes a la vez, ni más de una casilla en cada jugada.

Para descubrir la posición de este avión, y por ello decía que el radar estaba mermado en sus facultades, el jugador deberá proponer en cada jugada un radio (o distancia al punto A), y un sector angular que él cree corresponden a los del avión y cifrados como se indica en la figura 1 (la

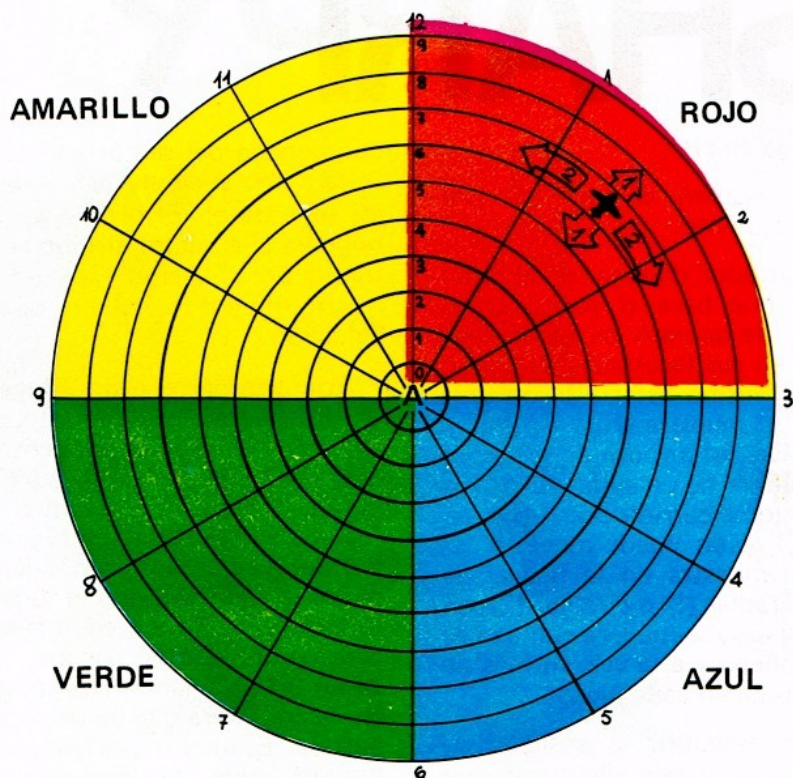
coordenada del sector angular se leerá como si de una esfera de reloj se tratase). El ordenador responderá si el sector elegido es correcto o no, y si el radio es correcto, mayor o menor, que el del avión.

Por otra parte, cada 3 jugadas el ordenador facilitará el color del sector en que se encuentra el avión, también cifrado, según la figura. La primera vez que será facilitado este dato será al principio del juego. Se dará por terminado el juego cuando el jugador localice el avión, en cuyo caso se considerará que ha ganado, o al cabo de 20 jugadas, tras las cuales se considera que ha perdido la partida. Es muy importante comprender que la posición del avión no es fija, y que en ello radica el interés del juego, en el que intervienen tanto la suerte como la habilidad del jugador, para determinar las posibles posiciones del avión gracias a las jugadas anteriores.

```

10: "G"E=ABS (10
  00*LOG (137/
  E)):E=E-INT
  E:X=INT (10*
  E)
20: RETURN
60: "A": CLEAR :
  INPUT "NUMER
  O INICIAL?";
  E:GOSUB 10:P
  =(X/6=INT (X
  /6))
70: GOSUB 10:IF
  (P=0)*(X=0)
  GOTO 70
75: IF (P=1)*(X>
  2)LET X=X-3*
  (INT (X/3))
80: N=10*P+ABS X
90: GOSUB 10:M=X
100: PAUSE " -":R
105: IF (R/3<>INT
  (R/3))THEN 1
  50
110: IF (N=12)+(N
  >=1)*(N<3)
  PAUSE "SECTO
  R ROJO"
120: IF (N>=5)*(N
  <6)PAUSE "SE
  CTO AZUL"
130: IF (N>=6)*(N
  <9)PAUSE "SE
  CTO VERDE"
140: IF (N>=9)*(N
  <11)PAUSE "S
  ECTOR AMARIL
  LO"
150: INPUT "SECTO
  R?";Y:INPUT
  "RADIO?";Z
160: IF Y=NPAUSE
  "SECTOR ";Y:
  " CORRECTO":
  GOTO 180
170: PAUSE "SECTO
  R ";Y:" INCO
  RRECTO"
180: IF Z=MPAUSE
  "RADIO ";Z:"
  CORRECTO":
  GOTO 200
185: IF Z>MPAUSE
  "RADIO ";Z:"
  DEMASIADO GR
  ANDE":GOTO 2
  00
186: PAUSE "RADIO
  ";Z:"INFERI
  OR AL REAL"
200: IF (Y=N)*(Z=
  M)PAUSE "GAN
  ASTE!";END
210: IF R>20PAUSE
  "PERDISTE.":
  END
220: R=R+1
230: GOSUB 10:P=(
  X/2=INT (X/2
  )):GOTO 240+
  30*P
240: GOSUB 10:N=N
  +1-2*(X/2=
  INT (X/2))
250: IF N>12LET N
  =N-12
260: IF N<1LET N=
  N+12
265: GOTO 100
270: GOSUB 10:M=M
  +1-2*(X/2=
  INT (X/2))
280: IF M>9LET M=
  M-1
290: IF M<0LET M=
  0:N=N+6
300: N=N-12*(N>12
  )
310: GOTO 100

```



La pantalla del radar.

La coordenada radial en este caso sería 7, y la coordenada de sector sería 1.

Figura 1.

El programa y su utilización.

Tanto para generar la posición inicial del avión como sus movimientos a lo largo de la partida, he utilizado una subrutina aparecida en la revista *COMPUTING TODAY*, en el volumen de Junio 1982, en el artículo "PC-1211 PROGRAMMING", y por ello hay que introducir un número positivo al principio de la partida.

Por lo importante que resulta conocer las jugadas anteriores, si las hubiese, para descubrir las posibles posiciones del avión, puede ser interesante ya apuntar éstas a lo largo de la partida, ya utilizar la impresora CE 122, para lo cual basta reemplazar los comandos PAUSE por PRINT: se facilita la lectura en el primer caso, y los datos aparecen en la banda de papel en la segunda.

Para iniciar (poner en marcha) el programa, pulsar SHIFT A en mode DEF.

Pablo Cuesta.

APROVECHANDO LA TECNOLOGIA DE CALIFORNIA SOFTWARE TOTALMENTE INTEGRADO

OPEN ACCESS

El nuevo Paquete de Software integrado para Aplicaciones profesionales en Microordenadores.

Con el programa Open Access, SPI ofrece un paquete que representa una nueva filosofía de Software de alta calidad. El concepto: Integración total de datos con guía óptima para el operador. Las posibilidades de aplicación: Como Banco de Datos, como programa de cálculos y proceso de textos, para la preparación y elaboración de datos de forma gráfica, como planificador de tiempo y como base de comunicación para otros sistemas de ordenadores.

Open Access ofrece al usuario un alto grado de funcionalidad junto con un diálogo cómodo con el usuario. Esto es señal de la fuerza de un sistema de programación altamente desarrollado y poderoso, esto facilita el trabajo, esto le permite concentrarse en resolver los problemas reales.

Open Access hace trabajar a su ordenador como Vd. quiere no al revés.

Naturalmente en Español.

Para esto necesita Open Access:

Decidir, organizar, planificar, administrar, coordinar, comunicar.

Esto ofrece Open Access:

Tratamiento de Ficheros, Cálculo, Gráficos, Proceso de Textos, Planificación del Tiempo, Comunicación.



SPI FORTALECE SU MICROORDENADOR



SPI SOFTWARE PRODUCTS INTERNATIONAL - REPRESENTACION EN ESPAÑA

C/ Profesor Waksman, 4-1º. izqda. - MADRID - 16 - TELFS. 458 04 00/07 50

Producto de matrices

Si se ha visto en la necesidad de efectuar muchos productos matriciales, se habrá dado cuenta de lo pesado de la tarea. No se preocupe, pues a partir de ahora su HP-41 lo hará por usted.

Como todos sabemos, el procedimiento para multiplicar matrices es el mostrado en el siguiente ejemplo:

$$\begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 \\ 3 \times 5 + 4 \times 7 \end{pmatrix} \quad \begin{pmatrix} 1 \times 6 + 2 \times 8 \\ 3 \times 6 + 4 \times 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

Para que dos matrices puedan ser multiplicadas, el número de columnas de la primera (que llamaremos A) ha de ser igual al número de filas de la segunda (B). El producto no es conmutativo. El resultado será una matriz (C) de un número de filas igual al de la primera y un número de columnas igual al de la segunda.

El programa listado nos permite realizar el producto de dos matrices A y B con una HP-41.

En el listado las E que aparecen pueden ser sustituidas por 1, y la E3 por 1E3. Han sido utilizados los exponentes cortos por ser más rápido que utilizar los números correspondientes.

El programa ocupa 234 bytes y utiliza 11 registros de datos, con lo que en una CV quedan libres 275 registros para el almacenamiento de las matrices. Esto permite por ejemplo el producto de una matriz de 9x10 por otra de 10x9.

El SIZE necesario es $NFA \times NCB + NFA \times NCA + NFB \times NCB + 11 = \text{dim C} + \text{dim A} + \text{dim B} + 11$

Los registros 00-10 son usados por el programa para varios propósitos.

101-126 Entrada de los elementos de las dos matrices A y B a multiplicar

127-131 Rutina general para inicializar el registro índice de los bucles.

Ejecución

Ejecutar PRDM. El programa nos pide $NFA \uparrow NCA$, introducir el número de filas de A, pulsar ENTER $\uparrow$, nú-

Al comienzo de la rutina de cálculo (LBL 05), su contenido es el siguiente:

01 $\equiv 1 + NFA/1000$
02 y 03 $\equiv 1 + NCA/1000$
04 y 09 $\equiv 1 + NCB/1000$

05 y 07 indican el registro del primer elemento de la matriz A
03 y 08 lo mismo para la B
06 y 10 para el primero de la C

mero de columnas de A y R/S lo mismo para la matriz B. A continuación nos pedirá $A1,1?$, primer elemento de la matriz A, (el primer número indica fila y el segundo columna), lo introduciremos y R/S, nos pedirá $A1,2?$... sucesivamente. La forma de introducción es por filas. Lo mismo para la segunda matriz, B, sólo que en este caso la introducción se hará por columnas. Una vez introducido el último elemento, el producto será realizado y visualizado por filas en la forma $Ci,j =$. Para ver cada uno de los elementos, pulsar R/S hasta el mensaje de FIN. Para efectuar un nuevo producto pulsar R/S

Modificaciones

Si queremos hacer productos en cadena habrá que modificar el programa quitando el CLRG y haciendo una rutina que pase la matriz producto a la

Líneas 01-19 Entrada del número de filas y de columnas de cada una de las dos matrices y prueba de que $NCA = NFB$
20-48 Calcula los índices de los bucles de entrada de los elementos de las dos matrices y del registro conteniendo el primer elemento de cada una de ellas.
49-75 Rutina de cálculo del producto.
76-99 Salida de la matriz producto.

EJEMPLO DE EJECUCION

$$\begin{pmatrix} 1 & 1 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -1 \\ 2 & 0 & 3 \\ 1 & 7 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 15 & 6 \\ 0 & 6 & 3 \end{pmatrix}$$

```

01*LBL "PRD 44 STO 01 87 FIX 4
M" 45 RCL 04 88 ARCL IND
02 CF 29 46 STO 09 10
03*LBL 00 47 RCL 06 89 PROMPT
04 FIX 0 48 STO 10 90 E
05 CLRG 49*LBL 05 91 ST+ 10
06 "NFA↑NCA 50 RCL 07 92 ISG 04
" 51 STO 00 93 GTO 08
07 PROMPT 52*LBL 06 94 RCL 09
08 STO 02 53 RCL IND 95 STO 04
09 RDN 00 96 ISG 01
10 STO 01 54 RCL IND 97 GTO 08
11 "NFB↑NCB 98 "FIN"
" 05 99 PROMPT
12 PROMPT 55 * 100 GTO 00
13 STO 04 56 ST+ IND 101*LBL 01
14 RDN 06 102 "B"
15 STO 03 57 E 103 FS? 00
16 "NO MULT 58 ST+ 05 104 GTO 04
P." 59 ST+ 00 105 "A"
17 RCL 02 60 ISG 03 106 ARCL 01
18 X=Y? 61 GTO 06 107 "F,"
19 PROMPT 62 RCL 02 108 ARCL 02
20 11 63 STO 03 109*LBL 02
21 STO 00 64 E 110 "F=?"
22 STO 06 65 ST+ 06 111 PROMPT
23 STO 07 66 ISG 04 112 STO IND
24 RCL 01 67 GTO 05 06
25 XEQ 09 68 RCL 09 113 E
26 STO 01 69 STO 04 114 ST+ 06
27 STO 09 70 RCL 00 115 ISG 02
28 RCL 02 71 STO 07 116 GTO 01
29 XEQ 09 72 RCL 08 117 RCL 03
30 STO 02 73 STO 05 118 STO 02
31 STO 03 74 ISG 01 119 ISG 01
32 XEQ 01 75 GTO 06 120 GTO 01
33 RCL 06 76 RCL 01 121 RTN
34 STO 05 77 FRC 122*LBL 04
35 STO 08 78 ISG X 123 ARCL 02
36 RCL 04 79 STO 01 124 "F,"
37 XEQ 09 80*LBL 08 125 ARCL 01
38 STO 04 81 FIX 0 126 GTO 02
39 STO 01 82 "C" 127*LBL 09
40 SF 00 83 ARCL 01 128 E3
41 XEQ 01 84 "F," 129 /
42 CF 00 85 ARCL 04 130 ISG X
43 RCL 09 86 "F=" 131 END

```

matriz A, que borre las B y C y nos pida la nueva B.

Si por el contrario lo que queremos es poder realizar productos con dimensiones mayores, podría pensarse en lo siguiente: Introducir la matriz B, a continuación la primera fila de la ma-

triz A y obtener así la primera fila de la matriz producto. A continuación, la segunda fila, con lo que sería calculada la segunda fila del producto, almacenándola en los mismos registros que la primera ya calculada, y así sucesivamente. Esto haría que el número de re-

gistros de datos necesarios sería: NFBx NCB + NCA + NCB + el número de registros utilizados por el programa.

Si bien estas dos opciones se dejan como labor al lector que necesite de ellas.

J.A. Deza

```

XEQ "PRDM"
NFA↑NCA 2 ENTER↑
3 RUN
NFB↑NCB 3 ENTER↑
3 RUN
A1,1=? 1 RUN
A1,2=? 1 RUN
A1,3=? 2 RUN
A2,1=? -1 RUN
A2,2=? 0 RUN
A2,3=? 1 RUN
B1,1=? 1 RUN
B2,1=? 2 RUN
B3,1=? 1 RUN
B1,2=? 1 RUN
B2,2=? 0 RUN
B3,2=? 7 RUN
B1,3=? -1 RUN
B2,3=? 3 RUN
B3,3=? 2 RUN
C1,1=5.0000 RUN
C1,2=15.0000 RUN
C1,3=6.0000 RUN
C2,1=0.0000 RUN
C2,2=6.0000 RUN
C2,3=3.0000 RUN
FIN

```


50 Años de Soportes Magnéticos BASF 1934
1984
Programada al futuro



Si está perdido sitúese con un mapa y una calculadora de bolsillo

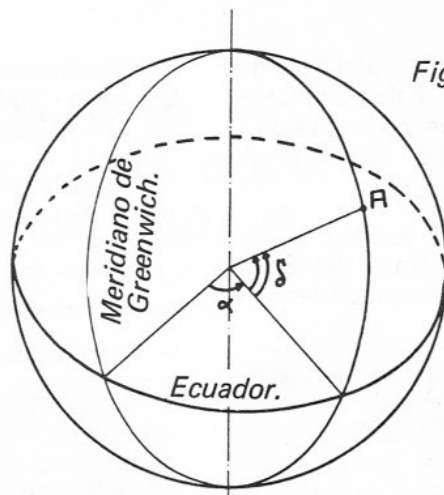


Fig. 1.

Hay que situarse.

Un pequeño programa para TI 58-59 que le permite situarse en este mundo. O cómo situarse con rapidez en un punto a partir de un mapa de carreteras ¿Se acabó el sextante? Juzguen ustedes.

Cuando la tierra era plana, la situación de un punto determinado era cosa fácil. Conociendo el centro del mundo, bastaba determinar la posición de ese punto con respecto al centro.

Pero los tiempos han cambiado. Desde que los australianos viven con la cabeza hacia abajo, no hay duda sobre la esfericidad de la tierra. A partir de este hecho, la localización de un punto necesita cierto aprendizaje.

Consideremos, por tanto, la Tierra como un esfera perfecta. (En realidad está ligeramente aplastada por los polos). Si se unen los dos polos por un semicírculo, se obtiene un meridiano.

El meridiano que pasa por Greenwich, está considerado, después de la Convención Internacional de 1.914, como meridiano origen. Con relación a éste, dado un punto A se determina la posición Este y Oeste, llamada longitud y representada por X. (fig. 1).

La longitud se expresa en grados, minutos y segundos. El perímetro de la Tierra (según un círculo perpendicular al eje que

atraviesa los polos), representa 360 grados.

1 grado = 60 minutos.

1 minuto = 60 segundos.

Por ejemplo, la longitud de Santiago de Compostela es $5^{\circ} 0' 10''$. Es decir, con un desplazamiento de 5° hacia el Oeste, a partir del meridiano de Greenwich, se alcanza el meridiano que pasa por Santiago de Compostela.

La posición Norte-Sur del punto A sobre su meridiano, está dada por la latitud. Es la medida del arco que una A con la intersección del meridiano de A con el Ecuador. La latitud se representa por S. (fig. 1).

Por ejemplo, la latitud de Santiago es $42^{\circ} 50' 04''$ Norte. Esto significa que Santiago se encuentra a $42^{\circ} 50' 04''$ al Norte del Ecuador.

Estas dos coordenadas bastan para determinar, sin ambigüedad, cualquier punto del globo.

Son utilizadas por topógrafos, geógrafos y navegantes.

Situarse, es determinar la posición —longitud y latitud— del lugar donde uno se encuentra.

Se admite, sin demostraciones, que se sabe, más o menos, dónde se encuentra. Es decir, que se puede situar sobre un mapa. (Lo sentimos por los lectores perdidos en una isla desierta).

Una llamada telefónica lleva a determinar las coordenadas, (Observatorio más próximo, ayuntamiento, catastro, etc.). Pero esto, no siempre, es la solución más rápida.

He aquí un método que permite obtener, de un modo bastante preciso, las coordenadas de un punto, cuando se conocen las de un punto cercano (algunos centenares de Kms.), y se dispone de un mapa (un mapa de carreteras, por ejemplo). Sabiendo que la longitud de un meridiano es 20.004,576 kms., y que ésta corresponde, del Polo Sur al Polo Norte, a una variación de latitud de 180° , se deduce la relación "Grado de latitud-kilómetro". (Ver recuadro).

Un kilómetro en la dirección Sur-Norte corresponde a una variación de latitud de,

$$\frac{180}{20.004,576}$$

grados decimales = 0,0089979413
grados decimales con 32 segundos. (Resultado obtenido con la función "INU ; D.MS").

Para determinar la longitud, todo esto se complicó un poco.

En efecto, si se clava una estaca en el polo Norte y se da una vuel-

ta alrededor de ella —algunos metros en total— se habrá realizado un recorrido que corresponde a una variación de longitud de 360° .

Pero si se da la vuelta a la Tierra siguiendo la línea del Ecuador, los 360° corresponderán a 40.000 Kms.

La variación de la longitud en función de la distancia recorrida en la dirección Oeste-Este, no se calcula, por tanto, con la ayuda de una simple regla de tres. Depende del valor de la latitud (S).

Con mayor precisión, la circunferencia de la tierra en el Ecuador es de 40.076,594 Kms. Un trayecto de un kilómetro en la dirección Oeste-Este, corresponde a una variación de longitud de

$$\frac{360}{40076,594 \times \cos S}$$

¿Un sextante?

No,
un programa.

Teniendo a nuestra disposición un método de cálculo infalible, sería deplorable volver al sistema prehistórico, lápiz + papel, para explicar nuestras fórmulas.

Algunos minutos para comprender el programa, algunos segundos para introducirlo y obtendrá sus grados con una precisión satisfactoria.

Un ejemplo:
Santiago - Valladolid.

Cálculo de las coordenadas geográficas de Valladolid, a partir de las de Santiago:

Sobre un mapa de carreteras, normal y corriente, de escala 1/500.000 (1 cm = 5 Kms) se han medido las siguientes distancias:

Valladolid se encuentra a 30 centímetros al Sur de Santiago y a 62 centímetros al Este. (Qué pequeño es el mundo).

Coordenadas geográficas de Santiago: Longitud $5^{\circ} 0' 10''$ Oeste; latitud $42^{\circ} 50' 04''$.

Método a seguir.

— 1 cm = 5 kilómetros, entonces
2 A

— Coordenadas geográficas de Santiago - 50010 B
425004 C

— Distancias medidas sobre el mapa de carreteras

+ 62 D
— 30 E

— Resultados. Pulsar A' y $x \rightleftharpoons t$

Del mismo mapa de carreteras, que dispone de coordenadas, deducimos que el resultado debe aproximarse a longitud $1^{\circ} 0' 7''$ - latitud $42^{\circ} 56' 00''$.

Como referencia para otros cálculos, puede ser útil BURGOS, por coincidir su longitud con la de Greenwich.

Modo de empleo

Una vez introducido el programa en memoria:

1 - Introducir n. Son los kilómetros que corresponden a 1 cm. sobre el mapa. n A.

2 - Introducir las coordenadas geográficas del punto P de referencia. Una longitud Este, llevará un signo "+". Una longitud Oeste, llevará un signo "-". Una latitud Norte llevará un signo "+", y una Sur, un signo "-".

Las coordenadas se introducen en forma de grados, minutos y segundos.

Longitud B ; latitud C.

Por ejemplo, si un punto tiene como coordenadas: longitud $4^{\circ} 27' 6''$ Oeste y la latitud es $26^{\circ} 6' 36''$ Norte, se introducirá:

— 4,2706 B ; 26,0636 C

3 - Introducir X. Es la distancia en cms., medida sobre el mapa, entre el punto P y el punto A, siguiendo la dirección Oeste-Este.

Algunas medidas útiles:

Radio ecuatorial: 6 378 386 m.
Radio polar: 6 356 912 m.
Radio mediario: 6 371 000 m.
Longitud de un meridiano (distancia entre 2 polos): 20004 576 m.
Correspondencia trayecto Norte-Sur/Variación de longitud: 1 km. representa una variación de latitud de $32''$.
10 Kms., $5'23''$.
100 Kms., $53'59''$.

Si el punto A se encuentra al Este del punto P de referencia, X será positivo. Si el punto A se encuentra al Oeste, X será negativo.

x D.

Así mismo, introducir y. Es la distancia en cms. medida sobre el mapa, entre el punto P y el punto A, siguiendo la dirección Sur-Norte.

Si A está al Norte de P, y será positivo. Si A está al Sur de P, y será negativo.

y E.

— Cálculo: A'.

Una vez terminada la ejecución, la calculadora visualiza la longitud de A. Pulsando sobre $X \rightleftharpoons t$, se obtendrá la latitud de A.

longitud X $\rightleftharpoons$ t latitud.

Para un nuevo cálculo, sólo habrá que introducir los valores que deben ser modificados.

Nota: Para una mayor precisión, es preferible determinar antes. Y (trazando un eje Norte-Sur que pase por P), y después X

000	76	LBL	008	01	01	016	14	D
001	11	A	009	92	RTN	017	42	STD
002	42	STD	010	76	LBL	018	03	03
003	00	00	011	13	C	019	92	RTN
004	92	RTN	012	42	STD	020	76	LBL
005	76	LBL	013	02	02	021	13	E
006	12	B	014	92	RTN	022	42	STD
007	42	STD	015	76	LBL	023	04	04



024	92	RTN	051	88	DMS	078	04	4
025	76	LRL	052	95	=	079	00	0
026	16	A*	053	42	STD	080	00	0
027	60	DEG	054	05	05	081	07	7
028	43	RCL	055	39	CD8	082	06	6
029	04	04	056	48	EXC	083	93	.
030	65	X	057	05	05	084	05	5
031	43	RCL	058	22	INV	085	09	9
032	00	00	059	88	DMS	086	04	4
033	65	X	060	58	FIX	087	55	+
034	01	1	061	04	04	088	43	RCL
035	08	8	062	52	EE	089	05	05
036	00	0	063	22	INV	090	85	+
037	55	+	064	52	EE	091	43	RCL
038	02	2	065	22	INV	092	01	01
039	00	0	066	58	FIX	093	88	DMS
040	00	0	067	32	XIT	094	95	=
041	00	0	068	43	RCL	095	22	INV
042	04	4	069	03	03	096	88	DMS
043	93	.	070	65	X	097	58	FIX
044	05	5	071	43	RCL	098	04	04
045	07	7	072	00	00	099	52	EE
046	06	6	073	65	X	100	22	INV
047	95	=	074	03	3	101	52	EE
048	85	+	075	06	6	102	22	INV
049	43	RCL	076	00	0	103	58	FIX
050	02	02	077	55	+	104	92	RTN

(trazando un eje Este-Oeste que pase por A). En efecto.

En efecto, para variaciones importantes de latitud, el valor de S está suficientemente modificado como para introducir: Un error no despreciable en el cálculo de la longitud.

En algunos mapas, los meridianos están dibujados. Se puede dar una cuenta de que la "recta" Norte - Sur presenta una ligera curvatura. Esta se sigue al desplazar la regla en el sentido de Y. Algunos milímetros, de más o menos, pueden convertirse en errores no despreciables en el valor de la longitud. La lástima es que la Tierra no haya sido plana.

Existe la posibilidad, de ahora en adelante, de determinar en algunos segundos, su nueva situación, gracias a este método tan original.

A saber si el correo le llegará a su dirección geográfica a partir de ahora.□

Antoine Jennet.

BASIÑOL: El Basic español

Tratando de demostrar que no es tan difícil como quieren hacernos creer los fabricantes e importadores de ordenadores, os presentamos en este artículo como hacer un Basic Español (BASIÑOL). Somos conscientes de no haber inventado nada, simplemente aportamos nuestro grano de arena. Los que disponeis de un Apple, adelante. Los demás, animo para hacer vuestras investigaciones al respecto. Nada es imposible.

Características generales del BASIÑOL

BASIÑOL es el BASIC APPLE-SOFT modificado de forma que las instrucciones —antes en inglés— aparezcan en Español y por lo tanto podamos efectivamente programar en un lenguaje Español. Aparte de esto, debo aclarar que la suerte o el concepto sobre el que reposa esta implementación nos va a permitir disponer de un traductor simultáneo entre el BASIÑOL y el APPLESOFT, esto es, que si escribimos un programa en BASIÑOL podremos luego visualizarlo en APPLESOFT —es decir en inglés— automáticamente además, claro está, de poder ejecutarlo. Lo mismo se puede decir si en vez de escribirlo en BASIÑOL lo escribimos en APPLESOFT: Podemos visualizarlo después en su versión Española.

Siguiendo con las características generales, merece la pena resaltar que **cualquier** programa previamente escrito en APPLESOFT podrá listarse y visualizarse en Español, pues esta adaptación no requiere ningún protocolo inicial que impida la traducción de programas anteriores. La compatibilidad con los programas BASIC es **total**.

Para aquellos que no conozcan el BASIC inglés pero en cambio deseen comprender un programa escrito en dicho idioma, bastará con que introduzcan el programa en su versión extranjera y luego por medio del traductor simultáneo lo visualicen y lo listen en BASIÑOL...

Conceptos técnicos previos

Si usted es eminentemente práctico, por favor siga en la próxima sección pues esto no le servirá de mucho.

El BASIC como todos sabemos está interpretado, esto es, las instrucciones se van interpretando, traduciendo y ejecutando a medida que se encuentran, y no de una vez por todas antes de ejecutarlo como ocurre en los compiladores.

```

JLIST
1  REM *****
2  REM *LISTADO 1.VISUALIZACION*
3  REM *DE LA MEMORIA.          *
4  REM *EN BUSCA DE LA TABLA...*
5  REM *****
6  REM
7  B = 13 * 4096: REM  HEX $D000

15 HOME
20 FOR I = 0 TO 15
25 FOR J = 0 TO 15
30 VTAB (I + 2): HTAB (J + 2)
35 U = PEEK (B + I * 16 + J)
100 GOSUB 1000: REM TEST ASCII
120 PRINT X$;: NEXT J: NEXT I
260 PRINT
269 INPUT "PULSE RETURN PARA SEG
    UIR";A$
270 B = B + 256
275 GOTO 15
400 END
1000 REM MIRA SI ES LETRA O CON
    TROL
1010 IF U < 128 THEN X$ = CHR$
    (U): NORMAL : RETURN
1020 IF U > 159 THEN X$ = " "
1040 IF U > 127 THEN X$ = CHR$
    (U): INVERSE : RETURN
  
```

LISTADO 1

Para traducir las instrucciones, el intérprete utiliza una codificación previa del programa de forma que todo resulte más rápido. Esta codificación es muy sencilla pues se trata únicamente de asignar un número a cada instrucción. Así, el listado en la memoria contiene una sucesión de números con los códigos —'tokens'— de cada instrucción BASIC del programa.

Esto quiere decir en definitiva que el lenguaje utilizado a nivel del intérprete no es ni inglés, ni español, ni chino: Es numérico.

¿Dónde aparece pues el conflicto lingüístico?

La respuesta es a nivel de edición. Es decir, el problema surge únicamente cuando queremos «ver» o «editar» o «escribir» un programa —un listado en definitiva—. En estos casos, el intérprete necesita presentar la información de una forma más agradable para la vista que una mera sucesión de números. Así pues convierte los números con las claves en palabras o instrucciones con las cuales se hallan relacionadas.

El procedimiento que se sigue es más o menos así: Se tiene una tabla en una zona de la memoria

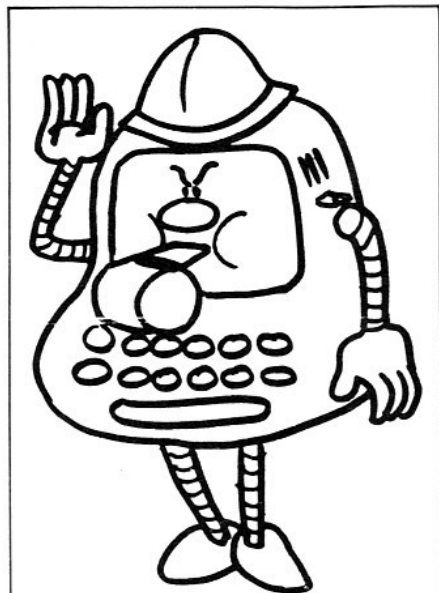
```

J
JLIST
1  REM *****
2  REM *LISTADO 2.          *
3  REM *LECTURA DEL BASIC *
4  REM *EN LA CARTA MEMORIA*
5  REM *****
6  HOME : PRINT
7  PRINT "PONGA EL SYSTEM MASTER
    EN EL DRIVE 1"
8  PRINT "PULSE RETURN PARA SEGUI
    R": INPUT A$
12 X = PEEK ( - 16255) + PEEK (
    - 16255): PRINT CHR$ (4)"B
    LOAD FPBASIC,D1":X = PEEK (
    - 16253) + PEEK ( - 16253)
  
```

LISTADO 2

con todas las palabras escritas una detrás de otra. Cuando encontramos que el código de una instrucción BASIC es «x», entonces cogemos la palabra número «x» de la tabla y la escribimos.

Para el caso de escribir un programa, el procedimiento es inverso. Dado una palabra, queremos hallar su código



```

1 REM *****
2 REM *LISTADO 3. *
3 REM *EDICION DE LA TABLA*
4 REM *****
5 PRINT "PULSE ESPACIO PARA NO M
  ODIFICAR UN CA-": PRINT "RAC
  TER DE LA TABLA.": PRINT " "-
  " PARA IR A LA POSICION ANTE
  RIOR": PRINT " " PARA IR A
  LA LINEA ANTERIOR"
6 PRINT " " PARA GRABAR EL BLOQ
  UE Y PASAR AL SIGUIENTE"
7 PRINT " " ANTES DEL ULTIMO CA
  RACTER DE UNA ": PRINT "PALA
  BRA"
8 PRINT " " PARA ACABAR"
9 PRINT : PRINT : INPUT
  "PULSE RETURN PARA EMPEZAR..
  ": AS
10 DIM BU(256)
14 B = 13 * 4096
15 HOME
20 FOR I = 0 TO 15
25 FOR J = 0 TO 15
30 VTAB (I + 2): HTAB (J + 2)
35 U = PEEK (B + I * 16 + J)
100 GOSUB 1000: REM TEST ASCII
120 PRINT X$: NEXT J: NEXT I
200 FOR I = 0 TO 15
210 FOR J = 0 TO 15
220 VTAB I + 2: HTAB J + 2: GET
  AS
250 GOSUB 2000: REM TEST ENTRAD
  A
255 PRINT AS
260 NEXT J,I
270 GOSUB 3000: GOTO 15
400 END
1000 BU(I * 16 + J) = U
1010 IF U < 128 THEN X$ = CHR$
  (U): NORMAL : RETURN
1020 IF U > 159 THEN X$ = " "
1040 IF U > 127 THEN X$ = CHR$
  (U): INVERSE : RETURN
2000 IF AS = ">" THEN GOSUB 300
  0: GOTO 15
2001 P = 16 * I + J
2005 IF AS = "." THEN POP : GOTO
  260
2006 IF AS = "*" THEN I = I - 1:
  POP : GOTO 260
2007 IF AS = "-" THEN J = J - 2:
  POP : GOTO 260
2008 IF AS = ":" THEN 5000
2010 IF AS = "/" THEN UL = 128: POP
  : GOTO 220
2015 IF ASC (AS) < 32 THEN UL =
  0: POP : GOTO 220
2020 IF ASC (AS) > 98 THEN UL =
  0: POP : GOTO 220
2022 NORMAL
2025 IF UL = 128 THEN INVERSE
2030 BU(I * 16 + J) = ASC (AS) +
  UL: UL = 0: RETURN
3000 FOR I = 0 TO 255: POKE B +
  I, BU(I): NEXT I: B = B + 255: RETURN
4000 FOR I = 0 TO 15: FOR J = 0 TO
  15: U = BU(I * 16 + J): GOSUB
  1000: PRINT X$:
4005 NEXT
4006 NEXT
5000 HOME : PRINT "PONGA EL DISC
  O DONDE VA A GRABAR EL": PRINT
  "BASINOL Y PULSE RETURN"
5002 GOSUB 3000
5003 NORMAL : PRINT
5005 PRINT "EL DISCO VA EN EL DR
  IVE 1": GET AS
5010 PRINT CHR$ (4) "BSAVE BASIN
  OL, A$D000, L$3FFF, D1"
5020 END

```

LISTADO 3

asociado para colocarlo en el programa codificado. Para ello buscamos en la tabla donde está colocada dicha instrucción y su posición nos dará el código.

Este es en términos generales el procedimiento seguido por un intérprete para traducir a números lo que se dice en palabras. Salta a la vista que para cambiar el idioma del BASIC solo tenemos que cambiar la tabla con las palabras asociadas a cada código de instrucción y modificarlas.

Para saber donde se encuentran hay que proceder con calma y tacto.

Lo mejor es hacer un pequeño programa en BASIC como el del listado 1 que nos permitirá traducir todos los números de la memoria a letras. Ya que los caracteres suelen codificarse en ASCII, tendremos grandes probabilidades de encontrar por

```

3 LIST
1 REM *****
2 REM *LISTADO 4. *
3 REM *LECTURA DEL BASINOL*
4 REM *EN LA CARTA MEMORIA*
5 REM *****
6 HOME : PRINT
7 PRINT "PONGA EL DISCO CON BASI
  NOL EN EL DRIVE 1"
8 PRINT "PULSE RETURN PARA SEGUI
  R": INPUT AS
12 X = PEEK (- 16255) + PEEK (
  - 16255): PRINT CHR$ (4) "B
  LOAD BASINOL, D1": X = PEEK (
  - 16253) + PEEK (- 16253)

```

LISTADO 4

algún lado una sucesión de números cuya traducción en caracteres sea la tabla de instrucciones. El único problema es saber por donde vamos a buscar...

Dado que según nos consta por los manuales el BASIC APPLE-SOFT se halla en la memoria ROM a partir de la dirección D000, será por esta dirección por donde empezaremos a buscar. Un último problema: Algunos de los códigos ASCII son controles, y a

```

oXewX\ YIcOsX_aL
s sdqJqTq$rir@r
Wsashv\vhwnwfwW!
w&wtwlrrrrrvrr
NrjYUr rZrJr s;s
sarEZ=Y YHYHxt
YjYLYmXkY gHX/X
czgTZ X$ViV [HV
k#l/k ^b TM_b
n.oAi oJoqo:p p
dgVfEc gefFfZf f
fy@gy)g( i{hj}
nPT_FN_On ^dd_
ENDFOENEXODATAIN
PUODEDIDREADGAT
EXOPRINOCALBPLO
OHLIOLVLIHGRGHG
HCOLOROHFLOODRAH
XDRAHTAHOMOROT
ESCALEOSHLOADTRA
CONOTRACONORMAB
NVERSEFLASHCOLOR
OPOVTAHMEMOLO
MEMOONERERESUMOR
ECALOSTOROSPEEDO
LEOGOTORUMIOREST
OROGOSUBRETURAR
ENSTODONWAILOLAD
SAVODEOPOKOPRINO
CONOLISTOCLEARGE
NEWTABOTOFNSFCOT
HENATONOSTEBEEOO
OANDONOOOSGINOA
BOUSAFRESERNOPDO
POSSQARNOLOEXOC
OOSINTANATOFEEOL
ENSTROVAOASOCHRO
LEFTORIGHTOMIDO

```

```

oXewX\ YIcOsX_aL
s sdqJqTq$rir@r
Wsashv\vhwnwfwW!
w&wtwlrrrrrvrr
NrjYUr rZrJr s;s
sarEZ=Y YHYHxt
YjYLYmXkY gHX/X
czgTZ X$ViV [HV
k#l/k ^b TM_b
n.oAi oJoqo:p p
dgVfEc gefFfZf f
fy@gy)g( i{hj}
nPT_FN_On ^dd_
FINDESDESIDATOP
EDIRBORDIDLEGGAT
XTOPRINOLLAMPLO
OHLIOLVLIHGRGHG
HCOLOROPTODIBUJA
XDIBOHTAPANOROT
EESCALELEEDIATRA
CONOTRACONORMAB
NVERSEFLASHCOLOR
OPOVTAHMEMOLO
MEMOONERERESUMOR
ECALBALAVELOCID
SEAIR ARUNSOREST
OROGOSUBVOLVERCM
NOSTOPEDESFALEED
GRADDEOPOKESCAC
ONOLISTOVACIACOG
NVOTABOAFNSPCOE
NTOANOOPASOEOO
OANDONOOOSGINOA
BOUSAFRESERNOPDO
POSSQARNOLOEXOC
OOSINTANATOFEEOL
ENSTROVAOASOCHRO
IZQUODERECOMEDO

```

FIGURA 1

Los caracteres con círculo van precedidos de /.

nosotros no nos interesa visualizarlos. Así pues por medio de un test del programa los convertimos en espacio».

Asimismo, los caracteres alfabéticos pueden tener dos códigos en ASCII, uno en el rango 193-218, y otro en el rango 65-90. Así por ejemplo a la letra A le corresponden el 193 o el 65 indistintamente. Así pues como esto puede resultar importante de cara a nuestras modificaciones, el programa los va a editar en inverso cuando el código sea mayor que 128 y en normal cuando sea menor.

Al ejecutar el programa, comienzan a aparecer en la pantalla un montón de caracteres sin sentido aparente lo cual es lógico dado que no todo lo que hay en memoria son frases y palabras. Sin embargo a partir de la posición de memoria D050 aproximadamente comienzan a desfilar las palabras clave del BASIC APPLESOFT... La suposición era pues exacta.

Un ligero vistazo y nos damos cuenta de que la última letra de cada palabra está escrita en modo inverso. Esto quiere decir que para señalar el final de la palabra el intérprete le asigna el código mayor que 128 —en realidad esto equivale únicamente sumar 128 al código del rango 65-90—.

Ahora ya sabemos como se codifican las instrucciones y donde. Sólo nos resta cambiarlo. Al llegar a este punto se nos plantea un problema o mejor varios problemas.

El primero —fundamental— es que el intérprete está en ROM, y por lo tanto no lo podemos modificar, pues no se puede escribir en este tipo de memorias. Para resolver este problema haremos lo mismo que se hace para utilizar el BASIC INTEGER. Lo implementaremos en la carta de lenguaje. Para ello tenemos que leer en la carta de extensión de memoria el BASIC APPLESOFT, lo cual se puede hacer con el programa del listado 2. (Al ejecutarlo hay que prever que en el drive se encuentre el SISTEM MASTER DOS /3.).

Una vez en la carta de lenguaje hacemos la conexión con esta de forma que aparentemente no habrá pasado nada ya que seguiremos con el mismo cursor en el borde lateral y el programa anterior no habrá desaparecido como cuando decíamos INT. La diferencia está en que ahora en BASIC lo tenemos en RAM listo para ser modificado.

Para modificar la tabla basta con editarla de nuevo, si bien en

la sección siguiente describimos un método alternativo más seguro para los que no quieran complicaciones.

Después de haber leído el BASIC APPLESOFT por medio del programa de listado 2, ejecutamos el programa del listado 3, el cual efectúa la conexión con la carta y edita la zona de memoria donde reside la tabla.

Después de visualizar las primeras 255 letras de la tabla, el programa se sitúa en el primer carácter para esperar nuestros cambios. El procedimiento de edición es bastante rudimentario dado que sólo lo utilizaremos una vez.

Basicamente el proceso a seguir es el siguiente:

*Si no queremos cambiar nada pulsamos '.' —.

*Si queremos colocar un

carácter en dicha posición lo pulsamos y listo.

*Si este carácter es el último de la palabra, entonces pulsamos previamente el carácter '/' seguido de la letra en cuestión que aparecerá en inverso en la pantalla.

*Si se equivoca puede retroceder a la posición anterior pulsando '—' siempre y cuando la posición anterior esté en la misma línea.

Si quiere situarse en la línea anterior entonces pulse ''.

En general debo aclarar que este procedimiento es muy delicado, ya que debemos usurpar más espacio del que originalmente había reservado para la tabla. En caso contrario podríamos borrar partes del programa intérprete lo cual tendría efectos desastrosos...

Lo que sí podemos —afortunadamente pues no estaba asegurado ni mucho menos— es cambiar la longitud de las palabras siempre y cuando la longitud total de la tabla no exceda la original. También hay que tener especial cuidado en que el orden de las nuevas instrucciones sea idéntico al anterior. Esto es, si en el original END precedía a FOR, en nuestra tabla FIN debe ir seguido de DESDE so pena de que el nuevo intérprete nos interprete todo al revés.

Para evitar líos, en la figura 1 se ve la tabla antes y después de la modificación para implantar el BASIÑOL. Así que bastará con seguir los mismos pasos sin olvidar que en las primeras posiciones que no se corresponden con caracteres, no debemos modificar nada —pulse espacio para seguir con la siguiente—.

Cómo implementar sin complicaciones su Basiñol

Dado lo delicado del método anterior, hemos pensado para usted uno más sencillo, pero más lento. En todo caso la seguridad es total.

En primer lugar escriba el programa del listado 2 y ejecútelo —habiéndolo insertado en el drive 1 el diskette SYSTEM MASTER DOS 3.3—. Ahora el BASIC APPLESOFT ha sido leído en la carta de lenguaje —por cierto usted lo tiene ¿no?— y puede ser modificado.

Ahora siga los pasos siguientes:

*CALL-151 (esto nos lleva al monitor. No tema.)

*D0D0.D25F

```
D0D0- 46 49 CE 44 45 53 44 C5
D0D8- 53 49 C7 44 41 54 CF 50
D0E0- 45 44 49 D2 42 4F D2 44
D0E8- 49 CD 4C 45 C5 47 D2 54
D0F0- 58 54 CF 50 52 A3 49 4E
D0F8- A3 4C 4C 41 CD 50 4C 4F
D100- D4 48 4C 49 CE 56 4C 49
D108- CE 47 52 41 B2 47 52 C1
D110- 48 43 4F 4C 4F 52 BD 50
D118- 54 CF 44 49 42 55 4A C1
D120- 58 44 49 42 D5 48 54 41
D128- C2 50 41 4E D4 52 4F 54
D130- BD 45 53 43 41 4C BD 4C
D138- 45 45 44 49 C2 54 52 41
D140- 43 C5 4E 4F 54 52 41 43
D148- C5 4E 4F 52 4D 41 CC 49
D150- 4E 56 45 52 53 CF 46 4C
D158- 41 53 C8 43 4F 4C 4F 52
D160- BD 50 4F D0 56 54 41 C2
D168- 4D 58 4D 45 4D BA 4D 4E
D170- 4D 45 4D BA 45 4E 45 52
D178- D2 52 45 53 55 4D C5 52
D180- 45 43 41 4C CC 41 4C CD
D188- 56 45 4C 4F 43 49 44 BD
D190- 53 45 C1 49 52 20 C1 52
D198- 55 CE 53 C9 52 45 53 54
D1A0- 4F 52 C5 A6 53 55 42 D2
D1A8- 56 4F 4C 56 45 D2 43 4D
D1B0- 4E D4 53 54 4F D0 45 CE
D1B8- 45 53 50 D2 4C 45 45 C4
D1C0- 47 52 41 C2 44 45 C6 50
D1C8- 4F 4B C5 45 53 43 D2 43
D1D0- 4F 4E D4 4C 49 53 D4 56
D1D8- 41 43 49 C1 43 4F 47 C5
D1E0- 4E 56 CF 54 41 42 A8 41
D1E8- CC 46 CE 53 50 43 A8 45
D1F0- 4E 54 C3 41 D4 4E 4F D4
D1F8- 50 41 53 CF AB AD AA AF
D200- DE 41 4E C4 4F D2 BE BD
D208- BC 53 47 CE 49 4E D4 41
D210- 42 D3 55 53 D2 46 52 C5
D218- 53 43 52 4E A8 50 44 CC
D220- 50 4F D3 53 51 D2 52 4E
D228- C4 4C 4F C7 45 58 D0 43
D230- 4F D3 53 49 CE 54 41 CE
D238- 41 54 CE 50 45 45 CB 4C
D240- 45 CE 53 54 52 A4 56 41
D248- CC 41 53 C3 43 48 52 A4
D250- 49 5A 51 55 A4 44 55 52
D258- 45 43 A4 4D 45 44 A4 00
```

LISTADO 5

*Ahora escriba el conjunto de instrucciones del listado 5.

Si tiene dudas sobre cómo se hace puede consultar el manual de referencia —página 44— o bien echar un vistazo a la figura 2

*D0D0:46 49 CE 44 45 53 44 C5
*D0DB:53 49 C7 44 41 54 CF 50
*D0E0:45 44 49 D2 42 4F D2 44
:

FIGURA 2

donde hemos simulado el comienzo de su tarea —no toda por piedad de nosotros...—.

*Una vez hecho esto ya tiene su BASIÑOL, pero ¡NO CORRA! Primero grábelo en el diskette (uno virgen preferiblemente) por medio de la instrucción siguiente:

BSAVE BASINOL, A\$DOOO, L\$3FFF

¡Y listo! Ahora ya puede ver su BASIÑOL sin más que pulsar Ctrl-C. Pruebe a decir lo siguiente:

DESDE X=1 AL 20 PASO 1: ESCR «YO SOY EL MEJOR»: SIG a ver qué pasa. Si le da syntax error es que lo hizo mal. Lo mejor es que se asegure de que ha escrito bien lo anterior, y si en efecto está bien seguramente siguió incorrectamente nuestras

instrucciones. Lo mejor es que lo repita todo desde el principio apagando primero el APPLE.

Si todo está en orden ya puede dormir tranquilo pues su BASIÑOL está grabado en diskette. Para entrar en BASIÑOL utilice el programa del listado 4. Usted puede inicializar los diskettes con dicho programa de forma que se ejecute al principio nada más conectar el APPLE. Para ello deberá tener en el mismo Diskette el programa BASINOL grabado previamente —puede copiarlo con el FID—.

Y nada más, en la tabla adjunta le damos el repertorio de instrucciones en BASIÑOL. Para traducir automáticamente un programa del BASIÑOL al BASIC pulse lo siguiente:

```
* x=peek(-16254)
* list
```

Con esta instrucción entra usted en el BASIC APPLESOFT por lo que debe escribir en inglés.

Para pasar del BASIC al BASIÑOL hacemos la operación inversa —siempre que el BASIÑOL estuviese ya en la carta de lenguaje—:

```
* x=peek(-16256)
* list
```

Con lo cual entramos en BASIÑOL y debemos escribir en español.

```
*
* LIST
1 CMNT *****
*
2 CMNT *LISTADO 1.VISUALIZACION
*
3 CMNT *DE LA MEMORIA.
*
4 CMNT *EN BUSCA DE LA TABLA...
*
5 CMNT *****
*
6 CMNT
7 B = 13 * 4096: CMNT HEX $D000

15 PANT
20 DESDE I = 0 AL 15
25 DESDE J = 0 AL 15
30 VTAB (I + 2): HTAB (J + 2)
35 U = PEEK (B + I * 16 + J)
100 SUBR 1000: CMNT TEST ASCII
120 ESCR X$: SIG J: SIG I
260 ESCR
269 PEDIR "PULSE RETURN PARA SEG
    UIR":A$
270 B = B + 256
275 IR A 15
400 FIN
1000 CMNT MIRA SI ES LETRA O CO
    NTROL
1010 SI U < 128 ENTC X$ = CHR$
    (U): NORMAL : VOLVER
1020 SI U > 159 ENTC X$ = " "
1040 SI U > 127 ENTC X$ = CHR$
    (U): INVERSO : VOLVER
```

Ejemplo de programa editado en BASIÑOL

Buen provecho y que se vean los progresos... □

Jaime Díez Medrano

apple COMPUTERS

TU NUEVA
TIENDA DE
MICROORDENADORES

C/ Fernández de los Ríos 36. Bajo dcha. Tel. 4 47 76 19 - Madrid-15

Introdúcese en la informática de forma racional. Tu puesto de trabajo puede depender de ello. CURSO GRATUITO DE BASIC
Curso de 20 horas, un ordenador por alumno y máximo de 9 alumnos por grupo. El precio del curso, 10.000 pts., te será descontado al comprar tu ordenador.

!! VEN QUEREMOS AYUDARTE !!

Ordenadores:

Spectravideo 318
Oric Atmos
NewBrain
Spectravideo 328
Toshiba T100
Toshiba T300

Impresoras:

NewPrint
Seikosha
C-Itoh
Toshiba

Discos para:

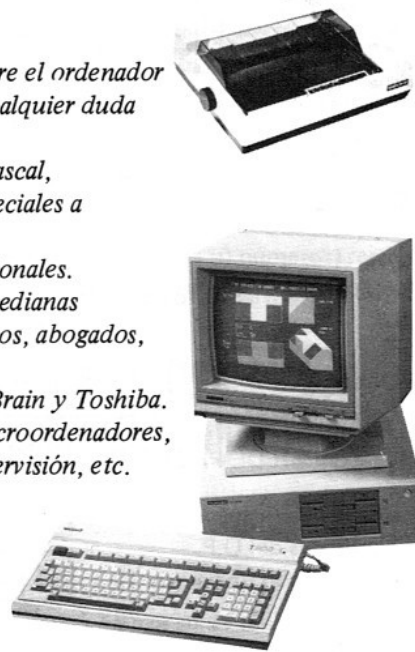
NewBrain
Spectravideo
Toshiba

Floppys

Monitores

Cassettes para ordenadores

- * *Técnicos especializados te asesorarán sobre el ordenador que se ajuste a tus necesidades y sobre cualquier duda que te surja.*
- * *Cursos periódicos y prácticos de Basic, Pascal, Ensamblador Z-80, CPM, etc. Cursos especiales a empresas, colegios, etc.*
- * *Programas de juegos, didácticos y profesionales.*
- * *Programas para comercios, pequeñas y medianas empresas, profesionales (dentistas, médicos, abogados, administradores, etc.)*
- * *Especialistas en programación para NewBrain y Toshiba.*
- * *Desarrollo de sistemas de control con microordenadores, (Medidas analógicas, telecontrol, telesupervisión, etc.)*



1ª OFERTA

5 números por 1.000 ptas.*

EL ORDENADOR PERSONAL



1 Presentación de ADAMICRO • Disquettes. Una tabla de índices para un acceso más directo • El Ordenador y la formación. Simulación y enseñanza asistida • Informática y Sociedad. La Cámara de Diputados • El TRS-80 modelo III en el banco de pruebas • El ordenador trata y mantiene la información en la consulta del médico • Pequeño glosario de informática • Utilización profesional • Juegos y Ordenador. Principios generales • Juegos: La huida con obstáculos • Perfeccionamiento. Para hacer buenos programas: una pizca de estructura y un puñado de módulos.

EL ORDENADOR PERSONAL



2 La exposición HARAMURI en Tokio • A pequeño comercio, pequeño ordenador • Abajo los prejuicios • Un tuno llamado VIC ronda bajo su ventana • Exploración anatómica y geográfica del ordenador • Banco de pruebas: PIPPLE II • Utilice un ordenador para la gestión de su club • Pequeño glosario de informática • ¿Está Ud. en forma mañana? • Iniciación a la programación • La arquitectura de los programas de juegos • Gestión familiar • El Apple pelado • ¿Recuerda el día de la semana en que nació? • Avanzadilla de pruebas: SINCLAIR ZX 81 • Las calcula-

doras programables también sirven para aprender • Las tablas de multiplicar.

EL ORDENADOR PERSONAL



3 Convierta las frías tablas en sugestivas curvas • Comencemos con la B con la A, BASIC • ¿Por qué una nueva informática? • Enseñe, al ordenador, Geografía • Iniciación: En la intimidad del 007 • Un servidor que sirve para todo • Los subprogramas • Avanzadilla: Sinclair (Continuación) • El juego del ahorcado • Avanzadilla: Sinclair (Continuación) • Banco de pruebas: CBM 8000 • La Dietética asesora da por calculadora • Encantos del Sharp • Marcador automático con Sharp 1500 • Las Vegas • Gran Premio de Penches • Pequeña música informática.

EL ORDENADOR PERSONAL



4 El sueño de una noche de invierno. Los 12 trabajos del microprocesador • Ensambladores, compiladores, intérpretes. La historia verdadera de su nacimiento • Las quinielas, relaciones de equivalencia • ¿Qué periféricos conectar a su ordenador? • La informática personal en Japón • Si está perdido, sítese con un mapa y una calculadora de bolsillo • Banco de Pruebas: Philips P2000 • Una cuestión de método. La programación estructurada • Tres novedades Sony • Avanzadilla de pruebas:

TRS80 color de Tandy Radio Shack • Enseñanza: A sumar se ha dicho • Dígalos con flores: Una tesis doctoral • Pasatiempos aritméticos: Los cuadrados mágicos o los crucigramas de la aritmética • Club de usuarios ZX81 Sinclair • Cosillas del ZX80 y 81.

EL ORDENADOR PERSONAL



5 Renta 1981 ó como calcular el impuesto • Seguimiento de los mundiales • Las quinielas. El método del potencial • Retrato de Familia • El futuro inmediato. Una vida diferente en la oficina La Ofimática • Banco de Pruebas: VIC-20 • Algunos consejos para una elección correcta de lógica de gestión de ficheros • Prueba de periféricos • ¿Quiéren Uds. jugar a los juegos del ordenador personal? • Cómo remitir artículos para su publicación • Cosillas del ZX81.



6 ¿El Basic le cansa? • Prueba de periféricos • Las quinielas (III). El símil cristalográfico • Ensayo para Los Angeles 84: 007 emite desde el Valle del Silicio • División de polinomios • Banco de Pruebas: Sharp MZ80-B • ¿Quiere Ud. jugar a los juegos del Ordenador Personal? • Los diskettes y su sistema de explotación • Cosillas del ZX 81 • Pequeña música informática.

EL ORDENADOR PERSONAL



8 San Francisco, siempre la más avanzada (la más hacia el Oeste) • Aplicación profesional: tres analistas de laboratorio • Tertulia de lenguajes. Un lenguaje potente: Forth • La generación de las pantallas planas • Como aprende morse con un Apple • Conectar un ordenador con un periférico: Los problemas de interface • Informática de bolsillo. Cambie de base cuando lo necesite con este programa para calculadora H.P. • Banco de Pruebas: BHP modelo 80-21 D Micral • ¿Quiere Ud. jugar a los juegos del Ordenador Personal? Material y Lógica • ¿Qué precauciones hay que tomar? • Juegos: Micro-Carambola • Superspy.

EL ORDENADOR PERSONAL



9 Programa Electoral: La Ley d'Hont • El programa supervisor coordinador de una circulación completa • Feria de los ordenadores • Música en el TRS-80 • Las quinielas (IV). Geometría dispersa • Las novedades de Texas: TI-88 y TI-57 LCD • Informática y medicina • Ponga un "zoom" en sus gráficas • Los juegos y el ordenador: Cómo programar una partida de dominó • Cuando el tren sigue la vía de su amo (I) • Banco de Pruebas: Atari 800 • Banco de Pruebas Logical: CORP. Generador de programas • Divulgación.

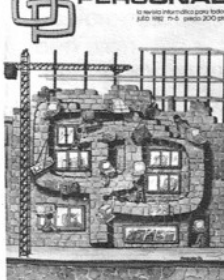
Un ordenador doméstico muy perfeccionado. El sistema YIS de Yamaha • Tertulia de lenguajes: Los lenguajes de programación de ordenadores • No descuide la seguridad de los programas y de los datos • ¿Qué es lo que da vueltas como un disco, es negro como un disco y es a la vez cuadrado? El Diskette • La caja negra • ¿Quiere Ud. programar los juegos del Ordenador Personal?

EL ORDENADOR PERSONAL



10 Houston: La NCC • ZX Spectrum • Ponga un ordenador en la máquina de escribir y consiga el tratamiento de textos • Respuestas del limón a la manzana • Banco de Pruebas: Olivetti M-20 • El Ordenador en casa: El juego del radar para Sharp PC-1211 • Las novedades del SICOB • Big-Pattern • Banco de Pruebas de Logical: Basi Data • Informática y Sociedad: ¿es de temer la informática? • Informática y Medicina: La informática ayuda al tratamiento del cáncer (1ª parte) • Juegue con el Ordenador Personal • Como ganar 140K octetos al menos taladrando un diskette • Las leyes de Golub del reino del ordenador • El encanto del Sharp • Informática de bolsillo: El tejano polaco.

EL ORDENADOR PERSONAL



11 Ecuaciones n-cuadradas • Basic y lenguaje má-

quina • Robots para jugar. Empezamos por una tortuga • Prepare un TRS para activar reles • Un ordenador que domina formas y colores • Ajedrez. Los principios • Banco de Pruebas: FACIT DTC 6522 • El laberinto de Candy. Juegos • Código de Barras, Impresora C. Itoh 8510 • Periféricos. Impresiones sobre impresoras • La cara oculta del Sharp • Avión Espía • Gestión de ficheros • La Informática ayuda al tratamiento del Cáncer 6 • ¿Es usted lógico? Un juego sobre HP-41C • Avanzadilla de pruebas: CASIO FX-702P • ¿Quiéren ustedes programar los juegos del Ordenador Personal?

EL ORDENADOR PERSONAL



12 Apple, IBM y Visicorp • Novedades en Japón • Lenguaje de programación ESCOLAR • Ciencia Ficción (mañana ¿qué ordenadores?) • Los juegos y el ordenador • Avanzadilla de pruebas: EL ACORN-ATOM • EL BASIC BASICO (La B con la A, Basic) • El tratamiento de textos en Japón • Y el Hombre creará el ROBOT (1ª parte) • Traductor - Monitor/Intérprete para MZ 80 B • Banco de Pruebas: EL NEWBRAIN • ELMASTER-MIND en Basic del ZX-81 • Programe en lenguaje máquina: ZX-81 - ROMPEMURROS • El secreto de los algoritmos calculadores • El microordenador en las clínicas • La función HIR de la TI 58/59.

EL ORDENADOR PERSONAL

13 Visita a la MICROFAIR • COMPEC-82: La informática Británica • Hewlett-Packard: Una estrategia diferente • Introducción al sistema CP/M • ¿Cómo seleccionar el software educativo? • Y el hombre creará el robot (parte II) • Avanzadilla de prueba: EL AIM 65/40 • LOGO • Carmela y la tortuga • LOGO • Meta una tortuga en su ZX-81 • Los nuevos antiguos contra los

BOLETIN DE PEDIDO

Sírvanse enviarme los números atrasados del ORDENADOR PERSONAL que marco con una equis ☒

Nombre Apellidos
Dirección Tfno.
Población D.P. Provincia

FORMA DE PAGO: ☐ Talón adjunto ☐ Giro Postal ☐ Contra reembolso

Deseo recibir los n^{os}

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13

Siguientes del O.P.

(marque con una equis ☒)

☒ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 20 ☐ 21 ☐ 22 ☐ 23 ☐ 24

(*) Esta oferta es válida sólo para cinco o más ejemplares, cada ejemplar de más se cobrará al mismo precio de 200 pts.

FIRMA:

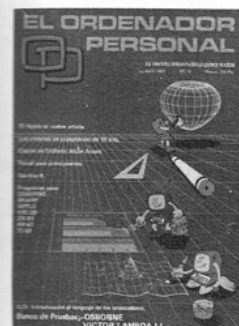
2ª
OFERTA

Ahorre 1.000 ptas. al suscribirse

nuevos modernos • Banco de Pruebas: EL SIRIUS-1 • Viaje alucinante a través del INTERPRETER EN UN MZ-80-B • Sistema periódico de los elementos. Gestión de ficheros secuencias en CBM 8032 • Los juegos y el Ordenador (Parte II). Cada vez

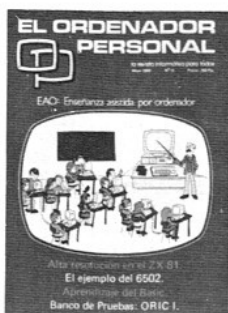


menos tiempo con el algoritmo Alfa-Beta • Minigolf de Karnak: para TRS y Video Genie • Las quinielas. Clases de equivalencias • Apple II • No juegues a las cerillas si no estás seguros de poder ganar HP-41 • Laberinto para MZ 80 B • Ficheros para Atom • Había una vez en el espacio intergaláctico del VIC-20.



14 Los Sistemas de Explotación 16 bits en guerra • ILO - Introducción al lenguaje de los Ordenadores • Ahorre memoria y aumente la velocidad de sus programas Basic interpretados • Banco de Pruebas: EL OSBORNE 1 • GENFRAS 8. Programa generador de frases para el ZX81 • Avanzadilla de pruebas: EL VICTOR LAMBDA II • Alerta. Las naves del Imperio contraatacan. ZX-81 • Recetario BASIC • PASCAL para principiantes • La informática y el diseño asistido: EL APPLE se vuelve artista-1e parte • Copia de Gráficos en alta resolución sobre impresora. ATOM-ACORN • Las confidencias del PC-1500 - 1ª parte

te • Un laberinto sin el hilo de Ariana - VIC 20 • Impresión de calendarios optimizada • HP 41.

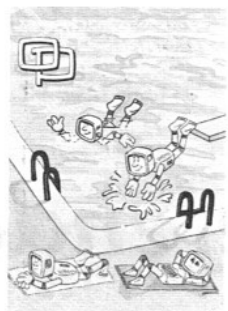


15 Diseño E A O • Médicos Consulten un O.P. • El Ordenador al servicio de las elecciones • Lenguaje máquina y ensamblador (el lenguaje del 6502) • Póngase Ud. al día • Por qué y cómo informatizarse... Consejos y recetas • Examinemos las memorias del Basic • Aprendizaje del Basic en un Instituto de bachillerato • Pascal para principiantes (2ª parte) • Banco de pruebas Basic • Confidencias del P.C. 1500 (2ª parte) • Banco de pruebas: ORIC-1 • El Acorn Atom protegiendo a la tierra frente a una terrible invasión • El Apple se vuelve artista (2ª parte) • El juego de Neisat • Métodos de Montecarlo (P.C. 1211) • Programa para alta resolución (Z X-81) • Activación y desactivación de sus aparatos domésticos • Como hacer un puente • Producto de Matrices (H P-41) • La astucia y la habilidad hacen más para fundar un club que la fuerza y los enfados.



16 El Cebit-83 de Hannover • Los ordenadores 16 Bits • Los procesadores 16 Bits • Banco de Pruebas: Dragón 32 • Los sistemas de explotación 16 bits • PAS-

CAL para principiantes (III) • Y el hombre creará el ROBOT (III) • Pánico en el fondo del mar • Las carreras de coches, un deporte de Salón • Programas de 1 K para el ZX-81 • Rally de Montecarlo para PC-1500 • Recetario Basic.



17 La 8ª West coast Computer Fair • Selección de equipos con fines educativos • El O.P. no hace al monje • Viaje al país de los juegos • Lenguaje máquina y ensamblador. El ejemplo del 6502 (y II) • 20.000 Leguas de viaje sub-pantalla • Periféricos HP-IL • Periféricos inteligentes para trabajar más rápidos • Sobre dos tipos de "Cracks" misteriosos en el ZX-81 • Primeros pasos del programa en notación algebraica • Síntesis musical • Nuevos usos para viejas calculadoras • ¿Conseguirá el ZX-81 salvar a los naufragos? • Conducir una locomotora • Un microbiólogo habla de sus ordenadores.



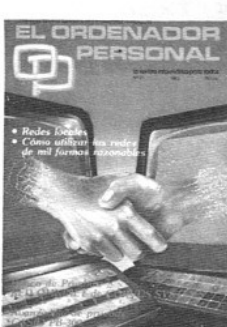
18 Memorización de pantalla en el PC-1500 • Pascal para principiantes (4ª parte) • Un sistema operativo estructurado. Unix • Un medidor de velocidad de cassette para Atom • Un poderoso programa para la correlación de sus datos. Parte I • Como transformar la impresora PC-100 en un trazador

de curvas • El Apple se vuelve artista (3ª parte) • Descubrir las artes gráficas gracias a la informática • El ordenador ayuda en la investigación de la paternidad • Recetario Basic • Las cuatro en raya del O.P. • Control informático del tratamiento antibiótico • ¿Quién pagará las cañas, usted o su HP-41? • Eche una carrera con su TI-59 • Y ante todo la música. Práctica de la síntesis musical • Los invasores han vuelto, yo los he encontrado • El ordenador jefe de estación (2ª parte)



20 SICOB-34 edición • Euromouse-83 • Repertorio de instrucciones del microprocesador Z-80 y lenguaje Assembler • Banco de Pruebas: KAYPRO II • Pascal para principiantes ficheros, procedimientos y funciones (5ª y última parte) • Banco de Pruebas programas: TIME ZONE: la máquina del tiempo • Vera Molnar o como dominar la casualidad • Creación artística • Banco de Pruebas: EL COMMODORE 64 • Dibujos en Perspectiva en su HP-41 • Le toca a Ud. ahora, fulminar al dragón del VIC-20 • El dibujo animado al alcance de su pantalla ZX81 • Lenguaje máquina para PC-1500 • Gráficos en el OSBORNE I • Cálculo de Velocidad de perfusión de fármacos vasoactivos • Juego de las siete y media para el CASIO FX 702-P • Integración por el método Simpson con ZX SPECTRUM.

21 Estudio detallado de un V.I.A. • Redes locales • Repertorio de instrucciones del microprocesador Z80 y lenguaje Assembler (2ª parte) • Banco de Pruebas: RED OMNINET • Cómo utilizar redes de mil formas razonables • Avanzadilla: Casio PB-300 • La falta de dinero no es tan grave -clubs- • Rutina en código máquina para proteger programas en BA-



SIC ZX81 • Supervisión de cuentas corrientes con un SHARP PC-1211 • Es el momento de esquivar HP 41 • Dump hexadecimal para PC-1500 • A hacer chuletas ZX-81 • Programas de 1 K para el ZX-81 • Tic-tac-tic-tac en Vic y en ORIC • Economice la ocupación de memoria TI-59.



22 Sistema operativo para microprocesador de 16 bits: MS-DOS • Repertorio de instrucciones del microprocesador Z 80 y lenguaje Assembler (3ª parte) • Viaje al centro del LOGO (3ª parte) • Un programa lleno de energía atómica. ATOM-ACORN • Rosas negras • Terrible amenaza a la federación galáctica HP-41 • Banco de Pruebas: MICRO PROFESSOR II • Geografía Espacial. PC 1.500 • Choque elástico. DRAGON 32 • Realizad nuevos caracteres en vuestro VIC-20 • Tratamiento de textos en la FX-702P • Estadística de dos variables para el ZX-81.

23 Viaje al centro del Logo (2ª parte) • Pequeños que casi no temen a los grandes • Realización de un protocolo RICHARDSON • ¿Quién es RICHARD Paul Jones? • Un laberinto sin el hilo de Ariadna. VIC-20 (Re-



sultado del concurso) • Ordenadores que ayudan a vivir • Espíritu ¿estás aquí? o los fantasmas del Commodore 64 (1ª parte). CBM-64 • La PC-1500 hace música. PC-1500 • BASINOL. El Basic español. APPLE • O.P. Defender. ZX SPECTRUM • ¿Dónde se encuentran los planetas? ZX-81 • Combinatoria. VIC-20.



24 Avanzadilla de Prueba: HP-150 • Viaje al centro del Logo (3ª parte) • Cuando el sueño se convierte en tecnología • Banco de Pruebas SORD M-5 • ¿Crecen de voz los O.P.? • Introducción al lenguaje de programación C • La PC-1500 aprende música • Escalera de color y escalofríos asegurados • Al claro de luna, amigo Pierrot préstame tu O.P. • Música es el arte de combinar sonidos • Cómo evaluar un biorritmo con la HP-41 • ¿Espíritu estás aquí? o los fantasmas del Commodore-64 (2ª parte) CBM-64 • Resolución de ecuaciones de 1er Grado • Rutina para formato de datos en impresión • Cuando el Atom tiene cita con la luna.

Deseo suscribirme a la revista EL ORDENADOR PERSONAL, por un año (10 números) recibiendo además LA GUIA de ORDENADORES PERSONALES y los dos números atrasados que marco a continuación.

MARQUE CON UNA ☒ LOS DOS N^{OS} ATRASADOS QUE DESEA RECIBIR.

1	2	3	4	5	6	8	9	10	11	12	13
14	15	16	17	18	20	21	22	23	24		

Nombre Apellidos
 Dirección Tfno.
 Población D.P. Provincia

Forma de Pago: ☐ Cheque adjunto ☐ Reembolso ☐ Giro Postal.

O.P. Defender

El presente programa es un juego del tipo Defender, bien conocido por Uds. pues fue uno de los más populares antes de que las tragaperras sustituyeran a las máquinas de marcianos en la mayoría de los locales públicos.

El programa está estructurado para que al ser todo en Basic se ejecute lo más rápidamente posible. La rutina principal se localiza al principio del programa para facilitar la pronta ejecución de los GO TO y GOSUB.

El resto de las rutinas de menor importancia como las instrucciones, etc.; están situadas al final de programa, ya que no requieren gráficos de movimientos rápidos.

La estructuración también permite que el juego sea fácilmente modificado y mejorado. Así mismo ciertas de sus rutinas pueden ser de interés para sus programas.

La finalidad del juego se detalla en las instrucciones; a continuación una explicación de cada una de las secciones del programa:

- 110** Llamada a la subrutina de la línea 9500 que define caracteres, etc.
- 120** Llamada a la subrutina de la línea 9000 para imprimir las instrucciones.
- 130** Llamada a la subrutina de la línea 8000, la cual permite establecer el nivel del juego.
- 140** Llamada a la línea 7000 donde se encuentra la su-

rutina de asignación de variables (puntuación inicial, posición, etc.)

- 150** Llamada a la subrutina en la línea 6000 que imprime la pantalla, el marcador, naves. N.º de seres, etc.
- 160** Llamada a la subrutina de la línea 5000 en adelante, que ejecuta la sintonía de introducción.
- 170** Llamada a la subrutina en la línea 1000; RUTINA PRINCIPAL.
- 180** Llama a la subrutina de la línea 4000, que informa al jugador del final de la partida y actualiza la tabla de máximas puntuaciones si corresponde.
- 190** Ordena que se repita el proceso entero (de la 130 a la 180).
- 500-590** Fija las variables del invasor.
 - L.A.=posición horizontal en la pantalla.
 - L.D.=Posición vertical.
 - A.I.=Decremento de la posición horizontal del invasor. (Esta determina la velocidad con que se mueve este a

través de la pantalla y es proporcional al nivel del juego.

D.I.= Incremento o decremento de la posición vertical del invasor. (Se elige aleatoriamente de la matriz D() que contiene los posibles incrementos o decrementos).

1000-1900 EL JUEGO:

1010 Llamada a la rutina del invasor e imprime este en su posición inicial.

1100 Borrado de la nave Defender de su anterior posición y la imprime en la nueva. C\$() contiene las dos posibles imágenes de la nave; cambiando C de 0 a 1 alternativamente para dar el efecto gráfico del chorro de los motores de propulsión. Se examina el teclado, si está pulsada la tecla de fuego y el invasor esta enfrente de usted, entonces se dibuja el trazo de un disparo laser delante de su nave y seguidamente se borra éste. Si su posición vertical es idéntica a la del invasor se llama a la rutina gráfica correspondiente.

1110 Se genera un número aleatorio, si éste es menor que su nivel de juego dividido por diez y el invasor está delante de Ud., se dibuja un disparo


```

10 REM DEFENDER
20 REM
30 REM EL ORDENADOR PERSONAL
40 REM
50 REM AUTORES M.L. Y J. MAURIN
60 REM
90 CLEAR 63999: REM PARA CARAC
TERES
100 REM Rutina de control
110 GO SUB 9500: REM INICIACION
120 GO SUB 9000: REM INSTRUCCIO
NES
130 GO SUB 8000: REM 5. NIVELES
140 GO SUB 7000: REM VARIABLES
150 GO SUB 6000: REM PANTALLA
160 GO SUB 5000: REM PAUSA
170 GO SUB 1000: REM BUCLE PRIN
180 GO SUB 4000: REM FIN
190 GO TO 130: REM REPETICION
500 REM SUBRUTINAS INVASORES
510 LET LA=31: LET LD=INT (RND*
14)+5: LET AI=1+INT (RND*3)+(SK
.3)+(SK*.6)+(SK=.9): LET DI=D(IN
T (RND*6)+1)
590 RETURN
1000 REM Rutina principal
1010 GO SUB 510: PRINT AT LD,LA:
CHR$ 16+CHR$ 5;"<":
1100 PRINT AT OSD,1:" "": AT SD,
1:C$(C+1): LET C=NOT C: IF (IN 3
2766=251) AND LA>3 THEN PRINT AT
SD,4:" "": BEEP .004,10: BEEP .004,9:
PRINT AT SD,4:" "": IF SD=LD THEN GO SU
B 2500: GO TO 1000
1110 IF RND<SK AND LA>5 THEN PRI
NT AT LD,4:CHR$ 16+CHR$ 4:" "":
1120 LET LA=LA-4: IF LD=SD THEN GO SUB 3000:
GO TO SIGUIENTE
1120 LET OSD=SD: LET SD=SD+2*((I
N 65276=253 AND SD<19)-(IN 65022
=254 AND SD>5))
1125 LET B$=B$(2 TO )+B$(1): LET
A$=A$(2 TO )+A$(1): PRINT AT 20
,0:CHR$ 15+CHR$ 4:A$:B$
1130 PRINT AT LD,LA:" "": LET LA=
LA-AI: IF LA<0 THEN GO TO 1200
1140 IF LD+DI>19 OR LD+DI<5 THEN
LET DI=-DI
1150 LET LD=LD+DI: PRINT AT LD,L
A:CHR$ 16+CHR$ 5;"<": IF RND<.3
THEN LET DI=D(INT (RND*(6-(2 AND
SK<.3)))+1)
1155 IF (LA=2 OR LA=3) AND LD=OS
D THEN LET SD=OSD: GO SUB 3000:
GO TO SIGUIENTE
1160 GO TO 1100
1200 LET HUMANOS=HUMANOS-1: PRIN
T AT 2,27:"0" ( TO 2-LEN STR$ HUM
ANOS):HUMANOS: FOR X=20 TO 10 ST
EP -2: BEEP .005,X: NEXT X: IF N
OT HUMANOS THEN GO SUB 2000: GO
TO 1900
1210 GO TO 1000
1900 RETURN
2000 REM TOMA DE HUMANOS
2010 PRINT AT OSD,1:" "": BEEP .1
,-27: BEEP .1,-27: BEEP .1,-27:
BEEP .4,-29
2020 BEEP .1,-27: BEEP .1,-27: B
EEP .1,-27: BEEP .4,-30
2030 PRINT OVER 1: INK 0: PAPER
2: FLASH 1: AT 4,0,,,,,,,,,,,,,
2040 FOR X=30 TO 0 STEP -5: FOR
Y=1 TO 255 STEP 13: OUT 254,Y: B
EEP .01,X*2: OUT 254,X: NEXT Y:
NEXT X
2050 PRINT PAPER 0: AT 4,0,,,,,,,,
, INK 4,,,,,,,,,,,,, OVER 1
2060 LET Z=1: FOR X=1 TO 15: PRI
NT AT 20,0: INK 4:A$:B$:AT 20,0,
,,, PAUSE Z: BEEP .01,X: BEEP .
01,10-X: LET Z=Z+1: NEXT X: FOR
X=60 TO 0 STEP -5: BEEP .01,60-X
: BEEP .01,X: NEXT X
2070 PRINT AT 20,0,,,,,
2490 RETURN

```

```

2500 REM EXPLOSION DE INVASOR
2510 FOR Y=1 TO 3: FOR X=1 TO 4:
PRINT AT LD,LA: INK X;"00": B
EEP .005,X*10+5: NEXT X: NEXT Y
2520 PRINT AT LD,LA:" "": BEEP
.1,-40: PRINT AT LD,LA:" "":
2530 LET SC=SC+10+(20 AND NA)=11
)+(30 AND NA>21)+(20 AND NA)=31)
+(20 AND NA)=41: PRINT AT 2,6-L
EN STR$ SC:SC
2540 LET NA=NA+1: IF NA/10=INT (
NA/10) THEN LET SK=SK+.1 AND SK
<.9)
2900 RETURN
3000 REM EXPLOSION DEL DEFENDER
3010 PRINT AT SD,1:" "": FOR X=1
TO 9: PRINT AT SD,2: INK 2;" "":
BEEP .02,10: PRINT AT SD,2: INK
6: BRIGHT 0:" "": BEEP .01,13:
NEXT X
3020 PRINT AT SD,2: INK 5;" "":
BEEP .1,-38
3030 PRINT AT SD,2: INK 4;" "":
BEEP .1,-40
3040 PRINT AT SD,2: INK 6;" "":
BEEP .1,-42
3050 PRINT AT SD,2:" "": BEEP .2
,-44
3060 PRINT AT SD,2:" "": PRINT A
T LD,LA:" "":
3070 LET VIDAS=VIDAS-1: GO SUB 6
200
3080 IF NOT VIDAS THEN LET SIGUI
ENTE=1900: RETURN
3090 GO SUB 7200: GO SUB 6080: G
O SUB 5000: LET SIGUIENTE=1000
3900 RETURN
4000 REM Rutina de fin del juego
4010 PRINT AT 8,0, INK 5:"
FIN DE LA PARTIDA"
4020 FOR X=200 TO -50 STEP -1: B
EEP .005,X/3: NEXT X
4030 CLS: IF SC<=5(SSK) THEN GO
TO 4200
4035 FOR X=0 TO 2: PRINT AT X,0:
INK 3:" "": INK 5:" "": NE
XT X
4040 PRINT INK 3: AT 0,0:"
IUY BIEN"
4050 FOR X=1 TO 5: FOR Y=20 TO 5
0 STEP 5: BEEP .01,Y: BEEP .01,Y
-5: NEXT Y: NEXT X
4070 PRINT "": RESTORE: FOR X=1
TO 4: READ Y$: FOR Y=1 TO LEN Y
$: IF Y$(Y)=" " THEN PRINT " "":
GO TO 4090
4080 PRINT PAPER 1:Y$(Y): PAUSE
.5
4090 NEXT Y: PRINT "": NEXT X
4100 DATA "USTED SE HA MERECID
O FIGURAR " EN LA LISTA DE F
AMOSOS DEL O.P.DEF
ENDER " POR FAVOR, TECLEE SU NO
MBRE A CONTINUACION" DOCE CA
RACTERES COMO MAXIMO"
4110 POKE 23624,71: INPUT INK 6:
"SU NOMBRE": LINE Y$: IF Y$=""
OR LEN Y$>12 THEN BEEP .1,0: GO
TO 4110
4120 LET S$(SSK)=Y$: LET S(SSK)=
SC
4200 CLS: PRINT INK 2,Z$: LET Y
$=CHR$ 16+CHR$ 5+" "
4210 PRINT INK 6:"LISTA DE HERO
ES EN CADA NIVEL"
4220 PRINT Y$ INK 6:"NIVEL
JUGADOR TANTEO"
4230 PRINT Y$
4250 FOR X=1 TO 5: PRINT : PRINT
"X: INK X+2:" S$(X)
"X: INK X+2:" 0000" ( TO 5-LE
N STR$ S(X)):S(X): NEXT X
4260 PRINT Y$
4270 PAUSE 100
4280 FOR X=1 TO 10: FOR Y=1 TO 5
: PRINT AT 0,0: INK Y,Z$: NEXT Y
: NEXT X
4290 PRINT INK 2: AT 0,0,7*
4300 PAUSE 50
4310 PRINT AT 21,0: INK 3:"DEFE
NDER": INK 5:"DEL ORDENADOR PER
SONAL"
4320 PRINT #0:"QUIERE USTED JUGAR

```

```

R..DE..NUEVO?..S..N"
4330 LET I$=INKEY$: IF I$<>"S" A
ND I$<>"N" THEN GO TO 4330
4340 IF I$="S" THEN GO TO 4900
4350 STOP
4900 RETURN

```

5000 REM TONO

```

5010 BEEP .2,-50: BEEP .2,-26: B
EEP .4,-22
5020 BEEP .2,-26: BEEP .2,-22: B
EEP .4,-18
5030 BEEP .2,-22: BEEP .2,-18: B
EEP .8,-15
5040 PAUSE 50
5900 RETURN

```

6000 REM DIBUJO DE LA PANTALLA

```

6010 POKE 23693,71: BORDER 0: CL
S
6020 PLOT 0,144: DRAW 255,0: DRA
W 0,31: DRAW -255,0: DRAW 0,-31
6030 PRINT OVER 1, PAPER 1; AT 0,
0
6040 PAPER 8: PRINT INK 6; AT 1,1
,"PUNTOS"; AT 1,12; "NAVES"; AT 1,2
2; "LEK-TORES"
6050 PRINT AT 2,1; "0000" ( TO 5-L
EN STR$ SC); SC; AT 2,27; "0" ( TO 2
-LEN STR$ HUMANOS); HUMANOS
6060 PRINT AT 20,0; INK 4; A$; B$
6070 GO SUB 6200
6080 PRINT AT 50,2; C$(C+1,4 TO )
6090 RETURN

```

6200 REM SUB. NO. DE NAVES

```

6210 PRINT AT 2,12; " "; A
T 2,12; FOR X=1 TO VIDAS: PRINT
INK 3; " "; NEXT X
6900 RETURN

```

7000 REM FIJAR VARIABLES

```

7010 LET A$="

```

```

7020 LET B$="
7030 DIM C$(2,11): LET C$(1)=CHR
$ 16+CHR$ 3+"<" +CHR$ 16+CHR$ 6+C
HR$ 19+CHR$ 0+">" +CHR$ 16+CHR$
7: LET C$(2)=C$(1,1)+CHR$ 4+"<" +
C$(1,4 TO )
7100 LET C=1: LET SC=0: LET VIDA
S=3: LET HUMANOS=15
7110 LET NA=0
7200 LET SD=15: LET OSD=50
7900 RETURN

```

8000 REM SELECCION NIVEL JUEGO

```

8010 POKE 23693,40: BORDER 5: CL
S
8020 PRINT "ELIJA

```

SU ELIJA

```

8030 PRINT "

```

8040 PRINT "INK 1" I 1 - FACI

```

8050 GO SUB 7000
8060 LET I$=INKEY$: IF I$<"1" OR
I$>"5" THEN PRINT AT 20,0; A$; B$
: LET A$=A$(32)+A$( TO 31): LET
B$=B$(32)+B$( TO 31): BEEP .001,
20: GO TO 8060
8070 LET SK=VAL I$/10: LET SSK=V
AL I$
8900 RETURN

```

9000 REM INSTRUCCIONES

```

9010 POKE 23693,71: BORDER 0: CL
S
9020 PRINT AT 2,1; "EL ORDENADOR
PERSONAL PRESENTA:"
9030 LET Z$="

```

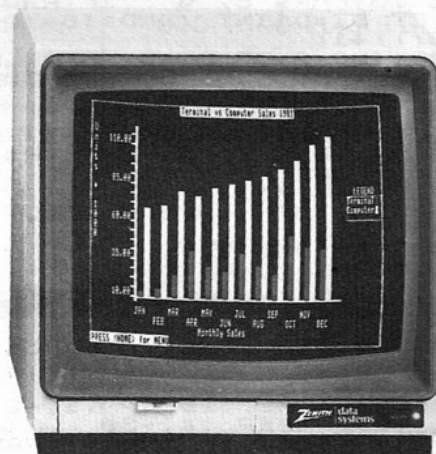
DER DEFEN



data systems



ORDENADOR PROFESIONAL con posibilidades de ampliación 6 y 16 bits — Microprocesadores 8085 y 8088 — 128K RAM ampliable a 768Kb. monitores monocromo o color — Uno o dos discos de 5,25" — opcional discos de 8" o Winchester interno o sea de 320Kb hasta 20Mb. Pantalla de 24 líneas y 80 columnas con definición de 225x640. Teclado de 108 teclas, con 13 de funciones especiales — 2 salidas serie R3 232 y 1 paralelo.



Modelo ZVM.122-E de 12" monocromo verde ambar
Modelo ZVM.134-E de 12" Color Alta resolución.
Modelo ZVM 124 de 12" monocromo compatible IBM.

Comercial CRUZ

MONTESA, 38 - Telfs.: 402 92 41 - 401 26 26 - MADRID-6


```

9040 PAUSE 100
9050 FOR X=5 TO 28 STEP 3: PRINT
AT 7,X-1; INK 6;Z$(X TO X+2);AT
8,X-1;Z$(X+32 TO X+34);AT 9,X-1
;Z$(X+64 TO X+66): BEEP .02,22:
PAUSE 20: NEXT X
9060 PAUSE 50
9080 FOR X=1 TO 10: FOR Y=1 TO 6
: PRINT AT 7,0; INK Y;Z$: BEEP .
01,-20: NEXT Y: BEEP .02,-38: NE
XT X
9100 PRINT AT 13,1:"PARA EL ZX S
PECTRUM DE AS.K."
9110 PAUSE 150
9120 LET Y$=" UD. ESTA AL MANDO
DE LA ULTIMA NAVE O.P.DEFENDER
DEL PLANETA MICRO SU MISION ES
ELIMINAR A TODAS LAS FUERZAS I
NVASORAS DE UNA GALAXIA LEJANA
QUE TRATAN DECAPTURAR A LOS PRE
CIADOS HABI- TANTES LLAMADOS LEK
-TORES PARA VENDERLOS COMO ESCL
AVOS.
9125 LET Y$=Y$+" SU NAVE E
STA EQUIPADA CON UN POTENTE LA
SER Y MOTORES IONICOS QUE LE PER
MITEN VOLAR SOBRE LAS MONTAÑAS D
EL PLANETA USTED DEBE DESTRUIR A
CADA UNO DE LOS IN-VASORES QUE
ENCUENTRE, DE LO CONTRARIO TOMA
RA UN LEK-TOR. SI DE-SAPARECEN T
ODOS, USTED SE DESIN-TEGARA.
9130 LET Y$=Y$+"
-CONTROLES DE LA NA
VE.
A- ARRIBA, Z- ABAJO, M-
FUEGO.
BUENA SUERTE!"
9140 POKE 23693,15: BORDER 1: CL
S
9150 FOR X=1 TO LEN Y$: IF Y$(X)
=" " THEN PRINT " "; GO TO 9170
9160 PRINT Y$(X);: BEEP .02,12
9170 NEXT X
9180 PAUSE 100: PRINT #0;"PULSE
ENTER PARA CONTINUAR"
9190 IF INKEY$<>CHR$ 13 THEN GO
TO 9190
9490 RETURN

```

PERE ES

```

9500 REM DATA DE CARACTERES
9510 POKE 23693,71: BORDER 0: PO
KE 23624,71: CLS
9520 PRINT " INK 2"
9530 PRINT INK 6;"VOY A CREAR CP
AFICOS-DEFINIDOS"
9540 POKE 23606,0: POKE 23607,24
9
9550 FOR X=15616 TO 16383: POKE
X+48384,PEEK X: NEXT X
9560 RESTORE 9600: FOR X=64128 T
O 64207: READ Y: POKE X,Y: NEXT
X
9570 FOR X=64264 TO 64471: READ
Y: POKE X,Y: NEXT X
9580 FOR X=USR "A" TO USR "U"+7
: READ Y: POKE X,Y: NEXT X
9590 REM DATA DE CARACTERES
9600 DATA 0,126,98,114,106,102,1
26,0

```

```

9605 DATA 0,56,24,Y,Y,Y,126,0
9610 DATA 0,126,6,126,96,Y,126,0
9615 DATA 0,126,6,126,6,Y,126,0
9620 DATA 0,14,22,38,70,126,6,0
9625 DATA 0,126,96,Y,126,6,126,0
9630 DATA 0,126,96,126,98,Y,126,
0
9635 DATA 0,126,6,6,12,24,48,0
9640 DATA 0,126,98,60,98,Y,126,0
9645 DATA 0,126,70,126,6,70,126,
0
9650 DATA 0,126,98,126,98,Y,Y,0
9655 DATA 0,126,98,124,98,Y,126,
0
9660 DATA 0,126,96,Y,Y,Y,126,0
9665 DATA 0,126,98,Y,Y,Y,126,0
9670 DATA 0,126,96,126,96,Y,126,
0
9675 DATA 0,126,96,126,96,Y,Y,0
9680 DATA 0,126,96,Y,110,98,126,
0
9685 DATA 0,98,Y,126,98,Y,Y,0
9690 DATA 0,24,Y,Y,Y,Y,Y,0
9695 DATA 0,6,Y,Y,Y,70,126,0
9700 DATA 0,100,Y,126,98,Y,Y,0
9705 DATA 0,96,Y,Y,Y,98,126,0
9710 DATA 0,126,106,Y,Y,Y,0
9715 DATA 0,122,106,Y,Y,Y,110,0
9720 DATA 0,126,70,Y,Y,Y,126,0
9725 DATA 0,126,98,126,96,Y,Y,0
9730 DATA 0,126,98,Y,106,Y,126,0
9735 DATA 0,126,98,124,98,Y,X,0
9740 DATA 0,126,96,126,6,Y,126,0
9745 DATA 0,126,24,Y,Y,Y,Y,6
9750 DATA 0,98,Y,Y,Y,Y,126,0
9755 DATA 0,98,Y,Y,Y,52,8,0
9760 DATA 0,106,Y,Y,Y,126,0
9765 DATA 0,66,102,24,Y,102,66,0
9770 DATA 0,98,Y,126,24,Y,Y,0
9775 DATA 0,126,6,12,24,48,126,0
9780 DATA 32,112,248,127,127,255
,255,48
9785 DATA 0,Y,Y,192,254,254,252,
0
9790 DATA 0,3,12,48,199,48,12,3
9795 DATA 0,Y,3,12,49,12,3,0
9800 DATA 0,Y,Y,Y,Y,85,0
9805 DATA 3,13,50,196,196,50,13,
3
9810 DATA 1,2,4,8,16,32,64,128
9815 DATA 128,64,32,16,8,4,2,1
9820 DATA 0,Y,Y,Y,Y,60,66,129
9825 DATA 0,Y,Y,24,Y,0,Y,Y
9830 DATA 0,Y,60,36,Y,60,0,Y
9835 DATA 0,126,66,Y,Y,Y,126,0
9840 DATA 255,129,Y,Y,Y,Y,255
9845 DATA 32,132,16,1,72,2,16,12
9
9850 DATA 0,Y,Y,3,Y,0,Y,Y
9855 DATA 0,Y,Y,192,Y,0,Y,Y
9860 DATA 0,Y,63,64,Y,63,0,Y
9865 DATA 0,Y,252,2,Y,252,0,Y
9870 DATA 3,12,112,129,Y,112,12,
3
9875 DATA 192,48,14,129,Y,14,48,
192
9880 DATA 0,Y,Y,Y,Y,119,0,Y
9890 DIM S$(5,12): DIM S(5): FOR
X=1 TO 5: LET S$(X)="ZX SPECTRU
M": NEXT X
9900 POKE 23658,6
9990 DIM D(6): LET D(1)=-2: LET
D(2)=2: LET D(3)=-1: LET D(4)=1:
LET D(5)=-3: LET D(6)=3
9999 RETURN

```

laser desde la nave invasora, (o sea el invasor está disparándonos). El disparo se borra y si la posición vertical del invasor es idéntica a la de Ud. entonces se llama a una subrutina gráfica que dibuja la explosión de su nave (lo siento). El ordenador salta después a la línea apropiada (contenida en la variable SIGUIENTE).

1120 Guarda la posición actual de la nave y la incrementa o decrementa de acuerdo a las teclas pulsada. También se

comprueba que la nave no se salga de la pantalla.

1125 Añade y desplaza las variables del paisaje A\$ y B\$, dando el efecto de avance al imprimir éstas en la parte inferior de la pantalla.

1130 Borra el invasor de la antigua posición y decrementa su situación horizontal. Si esta es menor que cero disminuye en uno el número de individuos y el control se va a la línea 1200.

1140 Se comprueba si la nueva posición vertical del invasor

está fuera de la pantalla; si lo está, se cambia el signo del incremento.

1150 Dibuja al invasor en su nueva posición y se genera un número aleatorio; si éste es menor que 0.3 entonces el incremento vertical del invasor se cambia.

1155 Se comprueba si la nave y el invasor ocupan la misma posición; en caso afirmativo se hace una llamada a una subrutina gráfica para la explosión de su nave.

1160 Repite la rutina principal

de la línea 1100 en adelante.

1200 Decrementa el número de individuos; si éste se agota llama a la subrutina gráfica para la explosión del planeta. Manda el control después a línea 1900.

1210 Repite la rutina principal de la línea 1000 en adelante.

1900 Devuelve el control al bucle principal en el principio del programa.

2000-2490 Subrutina que produce el efecto intermitente cuando se toman todos los seres del planeta.

2510-2520 Gráficos de la explosión de un invasor.

2530 Incremento del marcador. Este incremento es proporcional al número de invasores que usted ha eliminado anteriormente. Seguidamente se imprime el marcador.

2540 Incrementa en uno el número de invasores alcanzados. Si éste es múltiplo de diez se eleva un grado el nivel de juego.

2900 Devuelve el control al bucle principal.

3010-3060 Produce un efecto de explosión cuando Ud. es alcanzado por un invasor.

3070 Disminuye el número de naves y llama a la rutina de dibujo del número de naves disponibles.

3080 En el caso de que haya agotado el número de naves, hace SIGUIENTE igual a 1900 (un r.^o de línea) y devuelve el control al bucle principal.

3090-3900 Llamadas a las rutinas de variables, pantalla y sintonía. Además se devuelve el control al bucle principal.

4010-4020 Se informa al jugador de que ha terminado la partida.

4030 Se borra la pantalla y comprueba si se ha conseguido un tanteo récord en caso contrario devuelve el control a la línea 4200.

4035-4100 Se informa al jugador que ha obtenido un tanteo récord.

4110 Fija los atributos de las dos líneas inferiores de la pantalla para un INK blanco, PAPER (fondo) negro y Bright extra. Se pide al jugador que introduzca su nombre y se comprueba que no es demasiado largo o corto.

4120 Actualiza la tabla de puntuaciones récord.

4200-4270 Imprime dicha tabla.

4280-4290 Hace que la palabra DEFENDER cambie a diferentes colores.

4300-4310 Imprime el resto del mensaje después de un momento.

4320 Imprime en la última línea de la pantalla (usando #0); preguntando al jugador si desea otra partida.

4330 Espera hasta que se pulse «S» o «N».

4340 Si se pulsa «S» manda el control a la línea 4900.

4350 Se detiene la ejecución.

4900 Devuelve el control al bucle principal.

5000-5040 Ejecuta una pequeña sintonía para informar al jugador que está a punto de comenzar el juego.

5900 Devuelve el control al bucle principal.

6010 Pokea la variable del sistema ATTR P con los atributos de la pantalla, fija el color del borde y borra la pantalla.

6020-6030 Dibuja un rectángulo en la parte superior de la pantalla, con el fondo azul y el marco en blanco.

6040-6050 Imprime el tanteo, número de naves disponibles y número de seres; en el recuadro.

6060 Imprime el paisaje en la base de la pantalla.

6070 Llamada a la subrutina de impresión del número de naves disponibles.

6080 Dibuja la nave Defender en la posición inicial.

6090 Vuelta al bucle principal.

6210 Impresión gráfica del número de naves disponible.

6900 Devuelve el control a la parte correcta del programa.

7010-7020 Crea el paisaje en las variables A\$ y B\$.

7030 Crea la cadena C\$, que almacena las dos imágenes de la nave Defender, incluidos sus atributos (CHR\$ 16, etc).

7100-7200 Inicializa el marcador de puntos y las variables de posición.

C= (número de naves a imprimir)-1

SC= Marcador de puntos.

VIDAS= Número de naves disponibles.

HUMANOS= Número de seres que quedan.

NA= Número de invasores destruidos.

SD= Posición vertical de la nave Defender.

OSD= Antigua posición vertical de la nave.

7900 Devuelve el control a la parte apropiada del programa.

8010 Pokea la variable del sistema ATTR P con los atributos de la pantalla, fija el borde y borra la pantalla.

8020-8040 Pide al jugador que seleccione el nivel de juego.

8050 Llama a la subrutina de variables.

8060 Examina, para ver si una tecla de la 1 a la 5 está siendo pulsada. Si no, imprime el paisaje y lo mueve hacia la izquierda.

8070 Fija SK (Nivel del juego) al valor que ha sido pulsado dividido por diez; fija SSK igual a diez veces el valor de SK.

8900 Retorno al bucle principal.

9010-9110 Imprime el nombre del programa en letras grandes y la presentación.

9120-9180 Imprime las instrucciones a velocidad de lectura.

9190 Espera que se pulse ENTER.

9490 Retorno al bucle principal.

9510-9530 Fija los atributos de la pantalla y pide al futuro jugador que espere mientras se definen los caracteres gráficos y el nuevo juego de caracteres.

9540 Pokea la variable del sistema CHARS con 256 menos la dirección del nuevo juego de datos de caracteres.

9550 Copia los datos del juego normal de caracteres de la ROM a la RAM (encima del RAMTOP).

9560-9570 Pokea los datos para que los caracteres de la era espacial, se sitúen sobre el juego normal de caracteres en la RAM.

9580 Pokea en el área de gráficos definidos por el usuario: Invasores, nave Defender, etc.

9600-9880 DATAS para el nuevo juego de caracteres y gráficos definidos por el usuario.

9970 Crea la tabla de puntuaciones record.

9980 Pokea la variable del sistema FLAG 2 y el ordenador se pone en modo mayúsculas (CAPS SHIFT).

9990 Crea una matriz en direcciones, que contiene todas las direcciones para invasores.

9999 Retorna el control al bucle principal. □

Justo Maurín

INVASION ANTARTICA

**Proteja la vida animal en la antártica
Conviértase en un verdadero protector de los
animales**

Contenido:

Un pingüino blanco se defiende de un invasor mediante bloques de hielo azules (Mueve los bloques empujándolos con su cuerpo). El invasor no está dispuesto desde luego a que le ataquen. En todo momento está buscando la ocasión para atacar al pingüino. El juego termina en el momento que uno de los dos consigue alcanzar al contrario.

Cómo jugar:

Pon en marcha el programa y verás en pantalla 25 bloques de hielo azules, 12 de ancho por 11 de largo. Un invasor está alrededor de los bloques. Las ordenes para el juego son las cuatro flechas que indican las cuatro direcciones.

Pulsando cualquiera de ellas, conseguimos que el pingüino se desplace en la dirección indicada. Mueva el pingüino empujando a un bloque de hielo y éste se desplazará hacia el invasor en la misma dirección en la que se desplazó el pingüino. El bloque que hemos des-

plazado, seguirá su camino hasta encontrar otro bloque o bien el borde de la pantalla. El juego termina cuando o bien el invasor es aplastado por un bloque de hielo o el pingüino es alcanzado por el invasor. La puntuación aparece en la esquina superior derecha de la pantalla y se calcula restando los segundos que tardamos en alcanzar al invasor, partiendo de 100. Cuando el juego termina, pulse CTRL+Z y CTRL+T. Si desea volver a jugar pulse RUN.

Explicación del programa:

Puesto que es posible definir un total de 32 sprites, podemos en teoría utilizar 25 sprites para los 25 cubos. En la práctica, sin embargo, sólo podemos tener en una misma línea, un total de 4 sprites. Otra consideración es que si el pingüino o el invasor pasa por una línea que tiene los cuatro sprites, este se volvería invisible. A fin de evitar este inconveniente, todos los bloques de hielo son caracteres escritos con la instrucción PRINT y los sprites se utilizan sólo para el movimiento.

LISTA DE VARIABLES

XX,YY : Posición del pingüino.
D : Dirección de movimiento del pingüino.
P5, P6 : Dato de la tecla entrada.
BX, BY : Posición del invasor.
XO, YO : Distancia de movimiento de sprite.
SI, S : Cuida de que el pingüino o los bloques estén dentro de la pantalla.
X8, Y8 : Posición del sprite.
XA, YA : Coordenadas del pingüino o del hielo al moverse.
RXC, YC : Dirección del invasor.
TT : Puntuación.

MAPA DE PROGRAMA

10-50 : Creación de la pantalla.
60-110 : Entrada del teclado.
120-150 : Movimiento del pingüino.
160-220 : Movimiento de los bloques de hielo.
230 : Inicia o para la música.
240-270 : Subrutina para fijar la pantalla.
280-310 : Subrutina para mover los sprites del pingüino y de los bloques de hielo.
320-360 : Subrutina para mover el sprite del invasor.

LISTADO DEL PROGRAMA

```

20 print "1000":for I=0 to 132:vpoke I,0:next:I:let T=time
30 stchr "18343e3c7c3c382c" to 160:stchr "182c7c3c3e3c1c34"
to 161:scol 0,15:stchr "ffabd5abd5abd5ff" to 162:scol 1,5
40 stchr "24183c5aff182442" to 163:scod 2,163:scol 2,6:view
1,0,30,23:out 32,&B0
50 mag 1:let XX=96:let YY=80:let D=160:scod 0,D:scod 1,162:9
osub 240
60 let P5=inp(&35):let P6=inp(&36):if P5=2 or abs(XX-BX)<12
and abs(YY-BY)<12 then goto 230
70 gosub 340:if P5<>32 and P6<>4 and P6<>32 and P6<>64 then
goto 60
80 if P6=64 then let X0=2:let Y0=0:if E=161 then loc 0 to 30
0,300:let D=160
90 if P5=32 then let Y0=2:let X0=0
100 if P6=32 then let X0=-2:let Y0=0:if D=160 then loc D to
300,300:let D=160
110 if P6=4 then let Y0=-2:let X0=0
120 let S=0:let XA=XX:let YA=YY:gosub 280:let XX=XA:let YY=Y
A:let S1=S
130 scod 0,D:loc 0 to XX,YY:if abs(XX-BX)<12 and abs(YY-BY)<
12 then goto 230 else let X8=XX+8+8*X0/2
140 let Y8=YY+8+8*Y0/2:if XX=176 and P6=64 or YY>158 and P5=
32 then goto 60
150 if vpeek(Y8/16*12+X8/16)=0 then goto 60 else vpoke Y8/16
*12+X8/16,0
160 loc 1 to X8/16*16,Y8/16*16:Print cursor(X8/16*2,Y8/16*2)
;" ↓←← "
170 let S=0:let XA=X8:let YA=Y8:let X0=X0*4:let Y0=Y0*4:gosu
b 280:let X8=XA:let Y8=YA
180 let X0=X0/4:let Y0=Y0/4
190 if S=1 and S1=0 or P6=32 and XA=0 or P5=32 and YA=160 or
P6=64 and XA=176 or P6=4 and YA=0 then goto 220
200 if vpeek((Y8+4*Y0)/16*12+(X8+4*X0)/16)<>0 then goto 220
210 if S=1 then loc 1 to X8,Y8:goto 170 else loc 1 to X8-8,Y
8-8:goto 170
220 loc 1 to X8,Y8:vpoke Y0/16*12+X8/16,1:Print cursor(X8/16
*2,Y8/16*2):" ↓←←↑→":loc 1 to 300,300:goto 60
230 for I=1 to 9:for M=40 to 90:out 32,&A0+I:out 32,M:next:n
ext:out 32,&BF:Print "game over" end
240 stchr "ffffcccf3f3cccc" to 152,1:stchr "ffffcfcf3333cfc
f" to 153,1:stchr "f3f3cccf3f3ffff" to 154,1
250 stchr "3333cfcf3333ffff" to 155,1:vpoke &3B96,&50:for I=
1 to 25
260 let X=rnd(11):let Y=rnd(10):if vpeek(X+12*Y)then goto 26
0
270 vpoke X+12*Y,1:Print cursor(X*2,Y*2):" ↓←←↑→":next:retu
rn
280 gosub 340:if abs(XA-BX)>12 or abs(YA-BY)>12 then goto 30
0 else if abs(X0+Y0)<>2 then Print cursor(24,0):TT+C+C/2
290 goto 230
300 let C=C+1:let XA=XA+X0:let YA=YA+Y0:if P6=4 or P5=32 the
n goto 320
310 if XA<0 then let S=1:let XA=0:goto 330 else if XA>175 th
en let S=1:let XA=176:goto 330
320 if YA<0 then let S=A:let YA=0 else if YA>159 then let S=
1:let YA=160
330 return
340 let R=rnd(3):if R=0 then let XC=0:let XC=1 else if R=1 t
hen let YC=1:let XC=0
350 if R=2 then let YC=0:let XC=-1 else if R=3 then let YC=-
1:let XC=0
360 let BX=BX+8*XC:let BY=BY+8*YC:if BX<1 then let BX=0 else
if BX>175 then let BX=176
370 if BY<1 then let BY=0 else if BY>159 then let BY=160
380 loc 2 to BX,BY:let TT=time-T:Print cursor(24,0):TT+C:out
32,&A0+rnd(9):out 32,rnd(8):return

```


Para que su «MICRO ANDE»

SEIKOSHA

IMPRESORAS

GP-50 «LAS PEQUEÑAS»

30 c.p.s. y 46 columnas; 2 tipos de letras, cintas de varios colores. Impresora ideal para el aficionado o en aplicaciones donde se requiera una pequeña impresora numérica; con interface directa para ZX81 y SPECTRUM (S).

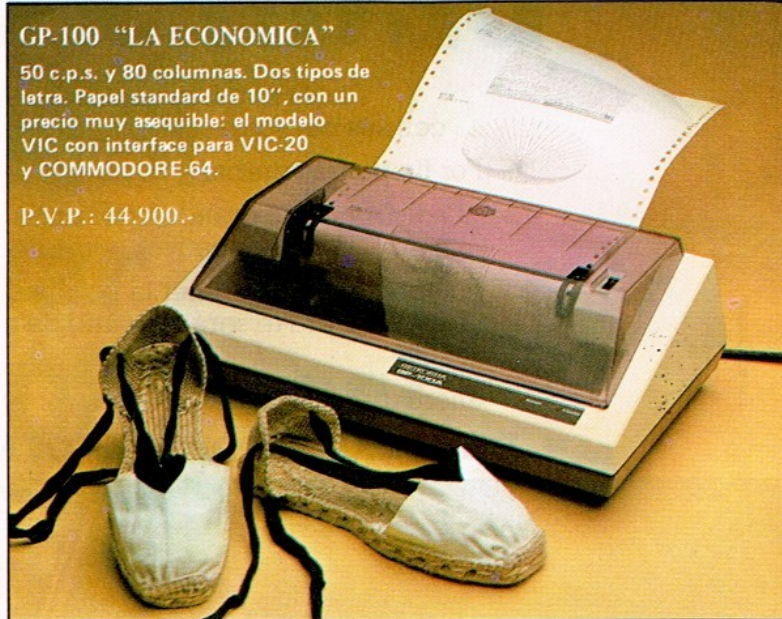
P.V.P. 24.900,-
(6.900,-) Pras.



GP-100 «LA ECONOMICA»

50 c.p.s. y 80 columnas. Dos tipos de letra. Papel standard de 10", con un precio muy asequible: el modelo VIC con interface para VIC-20 y COMMODORE-64.

P.V.P.: 44.900,-



GP-550 «LA STANDARD»

De 25 a 86 c.p.s. y de 80 a 136 columnas según el tipo de letra. Multitud de tipos de escritura con alta calidad de letra. Caracteres españoles. Fricción y acción. La mejor relación precio / prestaciones.

P.V.P.: 29.900,-



GP-700 «LA DE COLOR»

50 c.p.s. y 80 ó 106 columnas. 7 colores básicos; es capaz de mezclar punto a punto en una sola pasada, consiguiendo el Hard-Copy de cualquier microordenador con color.

P.V.P.: 98.500,-



GP-5420 «LA MAS RAPIDA»

120 c.p.s. y 104 en alta calidad de letra. De 136 a 272 columnas, con 15,5", 18 Kb. de buffer. Fricción y acción. Altas prestaciones, gran velocidad y alta calidad de letra.



P.V.P.: 299.000,-

Para cada modelo tenemos todos los números en cables, programas e interfaces; Paralelo Centronics, HP-IL, lazo de corriente, SINCLAIR, COMMODORE-64, VIC-20, IEE-448 (HP-IB), IBM y APPLE con Hardy-Copy en color, RS-232 con buffer, etc.

Si desea más información, llame o escriba a:

DiRAC S.L.

AV. BLASCO IBÁÑEZ, 114-116
TEL. 372 88 89 - VALENCIA-22
TELEX 62220

Delegación en Cataluña:
C/ Montaner, 60, 4, 1
Telf. 323 32 19
BARCELONA - 36

Deseo más información sobre la impresora:

Nombre _____
Empresa _____
Cargo _____
Dirección _____
Ciudad _____ Telf. _____

PONTE A LOS MANDOS DE UN SPECTRUM.

Ahora tu microordenador SPECTRUM es, aún, MAS con sus nuevos refuerzos: Microdrive, Interface 1, Interface 2...

¡Por fin podrás grabar y leer información de manera casi instantánea!

¡O disfrutar a lo grande con la más extensa variedad de programas tanto educativos como de mero entretenimiento!

Y sobre todo vas a tener la posibilidad de aprender a programar (que siempre te será muy útil) de una manera fácil y divertida.

No dejes pasar esta ocasión, ahora que puedes obtener mayor rendimiento de tu SPECTRUM.

Solicita información en la Red de Concesionarios Autorizados Investronica.

ESTE VERANO PONTE A LOS MANDOS DE UN SPECTRUM

J. M. PUBLICIDAD



IMPORTANTE:

Al adquirir los productos **SINCLAIR** exija la **TARJETA DE GARANTIA INVESTRONICA**, única válida en todo el territorio nacional y llave para cualquier resolución de duda o reparación. **INVESTRONICA** no prestará ningún servicio técnico a todos aquellos aparatos que carezcan de la correspondiente garantía.

DE VENTA EN CONCESIONARIOS AUTORIZADOS.

DISTRIBUIDOR EXCLUSIVO:
INVESTRONICA
CENTRAL COMERCIAL: Tomás Bretón, 60
Tel. 468 03 00 Telex: 23399 IYCO E Madrid.
DELEGACION CATALUÑA: Camp. 80 - Barcelona - 22

¿Conseguirá el ZX-81 salvar a los náufragos?

Si sólo dispone de 1 Ko en su ZX, es ampliamente suficiente para entrar el corto programa de juego que le proponemos. ¿Cuál? Esto depende de su estado de ánimo: se siente guerrero, y es un tanque que intenta dar en el blanco; se siente melancólico y manda un ramo de flores a su Dulcinea; quiere divertirse y... Sin lugar a dudas, sus distintos estados de ánimo le llevarán a realizar juegos inéditos. ¡Cuéntenoslos!

Su humor indica tormenta. ¡Cuidado, puede naufragar! Pero no importa mientras sepa utilizar las teclas "6"

y "7" para desplazarse —subir o bajar— y la tecla "5" para (supongamos) lanzar su salvavidas. Claro que —y desgra-

ciadamente— la desenfrenada oleada se lleva al pobre náufrago de manera aleatoria.

Tiene Vd. 20 salvavidas (variable N) y como la tempestad está causando grandes estragos por todos los lados, sólo dispone de ciento veinte (variable T) unidades de tiempo para recuperar a los náufragos (¡Ahora me acuerdo: Se trata del Titanic y con este frío glacial, qué desgracia!).

Su misión se acaba, por supuesto, cuando terminan las ciento veinte unidades de tiempo, o bien cuando se le agotan los veinte salvavidas. Entonces, el ZX le comunica el número de náufragos recogidos que podrán recuperarse de sus emociones.

La estructura del programa es sencilla: las líneas 100 a 170 corresponden a la fase de inicialización, el bucle principal va desde la línea 200 hasta la 360 y la línea 400 señala el final del programa.

Las principales variables son X (posición de su bote salvavidas), S (número de rescatados), B (posición aleatoria del naufrago) y AS\$ que consta de veintinueve rayitas —los veintinueve eslabones de la cuerda que tomó la precaución de anudar al salvavidas (¡No se le olvidó, claro!).

Claro que este programa resulta limitado pero está hecho con intención, lo cual le permite caber en 1 Ko.

Seguramente tendrá Vd. que añadir un poco de memoria para dibujar lo alto del iceberg, y no hablo de la parte escondida siquiera.

Michel Golay.

```

100 RAND
110 LET X=10
120 LET S=0
130 LET AS$=""
-----
140 LET A=INT (RND*20)
150 LET B=A
160 LET N=20
170 LET T=120
180 LET T=T-1
210 IF INKEY$="6" THEN LET X=X+
1
220 IF INKEY$="7" THEN LET X=X-
1
230 IF X=0 OR X=20 THEN LET X=A
240 CLS
250 PRINT AT X,29;"-";TAB 29;"
";AT B,A;" ";
260 IF INKEY$<>"5" THEN GOTO 32
0
270 LET N=N-1
280 PRINT AT X,A;AS$ ( TO 29-ABS
A)
290 IF X<>ABS B THEN GOTO 320
300 LET S=S+1
310 PRINT AT B,A;" ";
320 LET A=A+2-INT (RND*5)
330 LET B=B+2-INT (RND*5)
340 IF ABS A>23 THEN LET A=23
350 IF ABS B>18 THEN LET B=18
360 IF T AND N THEN GOTO 200
400 PRINT AT 9,9;"SCORE ";S;" ";
20-N

```



LA UNICA Y DEFINITIVA SOLUCION EN COLOR

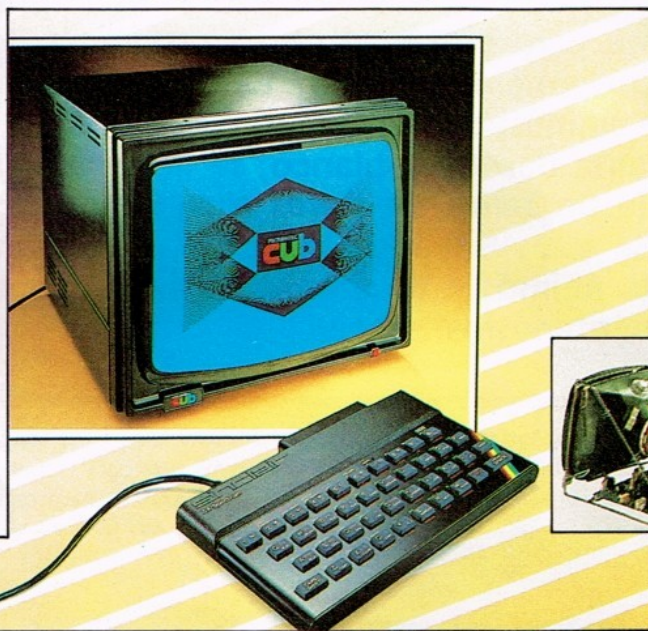
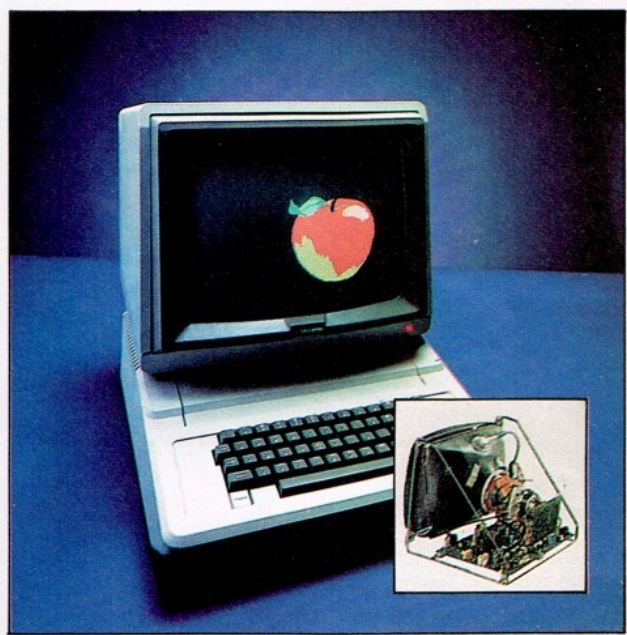
COMPATIBLE CON SU SISTEMA

1456/LI2. P.V.P.: 106.000

Modelos	Pixels
14" Standard	452 x 585
14" Media	653 x 585
14" Alta	895 x 585
20" Standard	505 x 585
20" Alta	860 x 625



1436/LS1. P.V.P.: 77.000



1431 MZ4. P.V.P.: 74.500

¿QUIEN NECESITA UN MONITOR EN COLOR?

Por supuesto toda persona que tenga un ordenador. Hasta ahora Vd. probablemente usaba su televisor doméstico con su ordenador y habrá notado bastantes interferencias, especialmente cuando visualiza textos. Los televisores no están básicamente diseñados para visualizar datos, ya que están contruidos con circuitos de codificación y modulación para aceptar únicamente las ondas de televisión a través del aire.

La diferencia entre su televisor y un monitor CUB, es que éste último está especialmente diseñado para la visualización de textos y gráficos, esto se evidencia inmediatamente en la imagen estable y clara que reduce notablemente el esfuerzo de la vista.

El monitor CUB está preparado para desarrollar las capacidades sofisticadas de visualización de los ordenadores de hoy y del mañana.

¿POR QUE ELEGIR UN MONITOR CUB?

Sólo la gama CUB de Microvitec, es suficientemente completa para cubrir la compatibilidad de casi todos los micro ordenadores del mercado.

Estos magníficos monitores británicos, son los únicos elegidos por el Gobierno inglés para usarlos en las escuelas primaria y secundaria de todo el país.

Nuestra gama de monitores de resoluciones standar, media y alta, más los modelos PAL/RGB, son compatibles totalmente con IBM PC/PCjr, APPLE II/IIe/III, SINCLAIR SPECTRUM/QL, COMMODORE 64/VIC 20, DRAGON 32/64, ORIC, BBC, ACORN ATOM, ATARI, ACT APRICOT, SHARP, IIT, TANDY, ADVANCE, CROMMENCO 501, LYNX, TEXAS INSTRUMENTS T-99/4A y muchos más.

Piense, cuando tome su decisión final, que sólo los CUB de Microvitec le pueden proporcionar la mayor calidad, rendimiento y fiabilidad al mejor precio.

TODOS LOS MONITORES CUB INCLUYEN:

- Garantía total por un año.
- Chasis aislado para máxima seguridad.
- Interruptor de potencia para un mejor rendimiento.
- Mínimo error de convergencia esencial para visualización de textos gráficos.
- Diseñados para introducir los standars reconocidos de seguridad (i.e. BS415).
- Chasis preparado para bajo consumo de potencia.
- Componentes de alta calidad para asegurar la máxima fiabilidad.
- Mando de conexión de potencia, plug y RGB.
- Diseño práctico, atractivo y moderno.
- Aprobación por la B.E.A.B. de nuestros más populares modelos.
- La mejor relación calidad-precio.
- La garantía de una gran firma como Microvitec que acaba de ganar el PREMIO REAL AL DESARROLLO TECNOLÓGICO 1984 EN INGLATERRA.

IMPORTADO Y DISTRIBUIDO EN
EXCLUSIVA PARA ESPAÑA:
multilogic

Ramón de Santillán, 15
Telf.: 458 74 75 - Madrid-16
Telex: 42710 FONOTXE

MICROVITEC
CUB

MONITORES COLOR



Realizad nuevos caracteres en vuestro VIC-20

Aquí tenéis algunas astucias para la programación de nuevos caracteres, que os permitirán utilizar las posibilidades de la alta resolución del VIC-20. Su principio es cargar una parte de la memoria viva con unos números adecuados, para generar caracteres, y esto en la versión de base con 3,5 KO de memoria viva.

El programa que os proponemos es un ejemplo de las posibilidades del VIC-20 en el dominio de la alta resolución.

No utiliza propiamente dicho la alta resolución, sino que simplifica la programación de nuevos caracteres. Esto merece algunas explicaciones.

El generador de caracteres del VIC-20 está en el momento de inicializar el ordenador, en memo-

VARIABLES DEL PROGRAMA

- A()* Tabla conteniendo los valores atribuidos a las líneas del carácter en construcción.
N Número de caracteres construidos.
I Línea en curso.
J Columna en curso.
AB Tecla apretada.
RB Respuesta a otro carácter.

ria muerta. Es posible, sin embargo colocarlo en memoria viva, accediendo a uno de los registros del circuito integrado encargado de la visualización del VIC-20.

Su dirección es la 36869.

Se puede hacer una pequeña experiencia muy simple si tecleáis POKE 36869,220 return, los caracteres se volverán dependientes del contenido de la memoria viva. Para volver al modo habitual, apretad simultáneamente las teclas RUN/STOP y RESTORE.

Se comprende en seguida que basta cambiar una parte de la memoria viva con los números ade-

cuados para obtener todos los caracteres que se quieran.

Tal y como están codificados los caracteres en el caso del VIC-20 están contenidos en una matriz de 8 x 8, es decir que se pueden obtener 2 a la 64 caracteres distintos, un número de 19 cifras. Como cada carácter está constituido de puntos, habrá que cambiar estos puntos para definir los caracteres.

Un punto encendido es un uno, un punto apagado es un cero

Para comprender la manera de codificar los puntos, tenemos que retroceder a la notación binaria. La notación binaria es lo que se llama en matemáticas una base, como la base 10 que es la que se utiliza normalmente. Esta base está caracterizada por no utilizar más que dos cifras, el 0 y el 1.

Se comprende cómo están codificados los puntos, un punto encendido es un 1, un punto apagado es un 0.

La matriz de caracteres está dividida en líneas y columnas, ocho en cada categoría. Cada línea está codificada por un número comprendido entre 0 y 255 ambos incluidos, esto es 256 posibilidades. Este número depende de los puntos encendidos en esta línea. Como lo hemos visto precedentemente, un punto encendido corresponde a un 1, pero este punto está igualmente caracterizado por su columna (o rango). Se calcula entonces la cifra de la línea como sigue: se atribuye a cada punto un número igual a 1 si está encendido, o a 0 si no lo está. Esta cifra se eleva en seguida a la potencia del rango, empezando en 0. Después se calcula el valor de cada línea, sumando los distintos valores atribuidos a cada punto que compone esta línea.

Se plantea en seguida el problema de saber dónde colocar este número.

En el generador de caracteres los caracteres están en el orden de su código de pantalla.

Cada carácter está codificado por la cadena de los ocho números representando las ocho líneas que le forman.

DESCRIPCION DEL PROGRAMA

- L.10* Dimensionamiento de AB, llamada para borrar la memoria.
L.15 Puesta a cero de A().
L.20 Orden de dibujar el casillero.
L.30 Posicionamiento del cursor.
L.40 Lectura de la tecla apretada.
L.50 Si es return se calcula el nuevo valor de la línea.
L.60 Si es un espacio se avanza el cursor.
L.70 Si estamos en fin de línea cambiarla.
L.80 Si estamos al final de un carácter, guardar el carácter.
L.90 Borrado del cursor.
L.110 Principio del bucle de almacenamiento de un carácter.
L.120 Carga de los valores correspondientes al carácter.
L. Principio del bucle de puesta a cero de la memoria viva de la casilla 6144 a 7680.
10000

READY.

```
10 DIMA(7):N=0:GOTO10000
15 FORX=0TO7:A(X)=0:NEXT
20 I=0:J=0:GOTO1500
30 POKE7680+22*I+J,160:POKE38400+22*I+J,6
40 GETA$
50 IFA$=CHR$(13)THENA(I)=A(I)+2*(7-J):POKE7680+22*I+J,160:J=J+1
60 IFA$=" "THENPOKE7680+22*I+J,32:J=J+1
70 IFJ>7THENI=I+1:J=0
80 IFI>7THENI10
90 POKE7680+22*I+J,32
100 GOTO30
110 FORX=0TO7
120 POKE6144+N*8+X,A(X)
130 NEXT
140 INPUT"OTRO CARACTER";R$
150 IFR$="SI"THENN=N+1:GOTO15
160 IFR$="NO"THEN180
170 GOTO140
180 PRINT"J":POKE36869,254:N=N+1
190 END
1500 PRINT"J"          L0
1510 PRINT"             L1
1520 PRINT"             L2
1530 PRINT"             L3
1540 PRINT"             L4
1550 PRINT"             L5
1560 PRINT"             L6
1570 PRINT"             L7
1580 PRINT"TTTTTTTT
1590 PRINT"01234567
1600 GOTO30
10000 FORX=6144TO7680
10010 POKEX,0:NEXT
10020 GOTO20
```

READY.

Un POKE 36839,254 coloca en generador de caracteres a partir de la posición 6144.

El primer carácter estará representado por el contenido de las posiciones de memoria siguientes 6144, 6145, 6145, 6147, 6148, 6149, 6150 y 6151.

Pasemos ahora al modo de empleo. Después de dar RUN, no preocuparse si no pasa nada durante algún tiempo. El programa vacía 1536 octetos de la memoria.

El casillero que aparece representa la constitución del carácter (8 por 8).

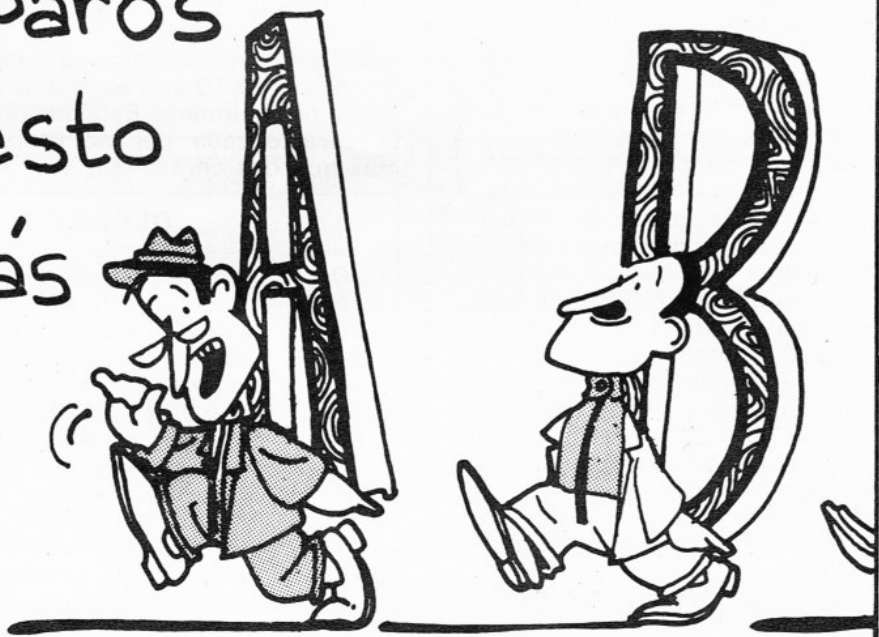
Precisemos que el primer carácter reemplaza la Ñ, el segundo la A, el tercero la B, etc.

Para formar un carácter hay que dar return para marcar un punto, y sobre la barra de espacios para saltárselo. Para probar el aspecto final de los caracteres, responder no a la pregunta de si queremos otro carácter, entonces el programa termina y podremos simplemente tecleando ver cómo quedan los caracteres. Para continuar basta con dar stop/restore y GOTO 15, con lo que podemos almacenar el siguiente carácter. Si queremos corregir un carácter ya pasado, antes del GOTO 15 dar N=X donde X es el código de pantalla a corregir.

Con este método podemos almacenar hasta 192 caracteres, pero es más que suficiente y en la práctica sólo estás limitado por tu imaginación.

Francisco Dervaux

No preocuparos
que el resto
viene detrás



16 K



GARANTIZAMOS EL CRECIMIENTO DE TU SPECTRUM

Si tienes un SINCLAIR ZX SPECTRUM de 16 K y deseas ampliarlo a 48 K ahora puedes hacerlo con toda **garantía**.

Acude a tu Concesionario Autorizado INVESTRONICA y en breve espacio de tiempo dispondrás de tu Spectrum con **MAS POTENCIA**.

Además INVESTRONICA ampliará por tres meses la garantía de tu equipo, independientemente de la fecha de adquisición y te obsequiará con una Cinta de Demostración de 48 K.

IMPORTANTE:

Al adquirir los productos **SINCLAIR** exige la **TARJETA DE GARANTIA INVESTRONICA**, única válida en todo el territorio nacional y llave para cualquier resolución de duda o reparación. INVESTRONICA no prestará ningún servicio técnico a todos aquellos aparatos que carezcan de la correspondiente garantía.

**DE VENTA EN CONCESIONARIOS
AUTORIZADOS.**



48 K

Amplía,
ahora, tu SPECTRUM
de 16 K a 48 K.

**Con garantía...
Con más garantía.**



**DISTRIBUIDOR
EXCLUSIVO:**

INVESTRONICA

CENTRAL COMERCIAL: Tomás Bretón, 60.
Tel. 468 03 00 Telex: 23399 IYCO E Madrid.
DELEGACION CATALUÑA: Camp, 80 - Barcelona - 22

TENER UN GRAN ORDENADOR NO CUESTA MAS

118.500 pts.

BASE 64A

Características BASE 64 A

RAM: 64 Kb libres usuario, ampliables hasta 192 Kb.

ROM: 32 Kb; 4 Kb para monitor, 18 Kb lenguaje BASIC, 10 Kb para editor de textos.

Teclado ASCII, tipo máquina de escribir 72 teclas con teclado numérico adicional.

Alta fiabilidad del teclado (diez millones de pulsaciones garantizadas).

Doble generador de caracteres: Americano y Español.

Instrucciones BASIC directas con una sola tecla.

Mayúsculas y minúsculas.

El más profesional
de su familia

Alta resolución gráfica: 280 x 192 puntos.

8 conectores para ampliaciones.

80 columnas, pal color, CP/M con Z-80, comunicaciones RS-232, etc.

15 colores.

Compatible con más de 10.000 programas

Unidad de disco flexible 5 1/4
almacena 143 Kb

* MONITOR 12" fósforo verde con base orientable de alta resolución, 33.000 ptas.

NOTA: Abrir la tapa no vulnera la garantía.



MICOMPSA

General Perón, 32. Madrid-20. Tel. 456 22 11

El juego del globo

Maniobra con habilidad, hágase con el máximo de puntos y no deje de vigilar su indicador de combustible.

Descripción del juego:

El objetivo del juego es capturar el máximo número de proyectiles blancos (que aportan puntos) esquivando los proyectiles negros, (que destruyen el globo) y posarse finalmente en el islote del que se partió.

El jugador deberá hacer despegar al globo y controlarlo gracias a unos quemadores (cuyo combustible es un gas del que conocemos la cantidad disponible en cada instante) y que se accionarán pulsando la tecla de espacio —teniendo en cuenta que el globo subirá cuantas veces se apriete dicha tecla (ojo a las teclas auto repetitivas) y bajará cuando haya cumplido el número de subidas que se le hayan indicado—. Este efecto de acumulación puede resultar molesto al no ser instantánea la respuesta al soltar la tecla de espacio, pero se corresponde con lo que ocurre en la realidad.

Por otra parte, el globo es arrastrado por la fuerza del viento que sopla con dirección variable en cada una de las tres zonas de la franja atmosférica a las que nuestro aerostato puede acceder, estando simbolizadas las direcciones del viento por flechas a un lado de la pantalla.

Variando pues la altura se podrá modificar también la velocidad horizontal, siendo recomendable mantenerse siempre centrado en la pantalla preveyendo un posible agotamiento

del combustible (partimos con quinientas unidades, y el ordenador nos avisa cuando nos quedan solo cien; un aporte especial de trescientas unidades nos será dado como premio cuando alcancemos los 1.500 primeros puntos, y más tarde los 3.000).

En caso de rebasar los límites derecho o izquierdo de la zona de juego el globo volverá a aparecer en el extremo opuesto de la pantalla. Alcanzando un tope de altura el programa impedirá seguir subiendo.

Descripción del programa

Como la mayoría de los video juegos este programa tiene una estructura de bucle en el que tras una inicialización de los parámetros y el dibujo de fondo se suceden.

— bucle general: recogida de datos del teclado y variación en memoria de la posición del globo

— llamada a la subrutina de dibujo del globo

— movimiento y creación de proyectiles

— llamada a la subrutina de impactos y puntos

— vuelta al principio, en caso de no terminar la partida.

El programa trabaja en modo texto (80 columnas, 25 filas) y utiliza caracteres gráficos.

Estos se van sumando a los contro-

les de cursor en cadenas de caracteres que se imprimen de una vez en pantalla, con lo que la imagen permanece clara y sin serpentos molestos. Para mejorar la velocidad del programa es conveniente traducir estas rutinas de dibujo a lenguaje de máquina.

La aleatoriedad del juego viene de la existencia del generador de números aleatorios (RND) que hemos inicializado tomando como parámetro para la función RANDOMIZE los segundos en el reloj que lleva incorporado la tarjeta de memoria.

El resultado es mejor que si introdujeramos otro parámetro, pero no es en absoluto necesaria esta sofisticación.

Por último haré a continuación un breve repaso de las variables y códigos ASCII utilizados.

- GB \$: dibujo del globo. Cadena de C. con controles de cursor.

- CX, CY, VX, VY coordenadas y velocidad del globo.

- GR1 (J) matriz columna de 5 elementos que contiene los colores de fondo, bajo forma de códigos ASCII los tres primeros son caracteres con tres tipos de punteado, y los dos otros blancos.

(Sentencia n.º 610).

- ASCII: 028: cursor a la derecha, 029: cursor a la izquierda, 030: cursor arriba, 031: cursor abajo.

Pablo Cuesta Diego

```

10 REM *****
20 REM **** JUEGO DEL GLOBO *****
25 REM *****
30 REM **** COPYRIGHT : El autor y *****
35 REM **** EL ORDENADOR PERSONAL *****
40 REM *****
50 REM-----Iniciación
60 CLEAR
70 DIM GR1(5)
80 DEF FNCOL(CY)=GR1(INT((CY-1)/5)+1):DEF FNVIEN(CY)=VV(INT((CY+7)/5))
90 DIM PX(6):DIM PY(6):DIM CP$(6):CLS:KEY OFF
100 CX1=35:CY=35:VX=0:VY=0:GAS=500:SCORE=0
110 T=VAL(RIGHT$(TIME$,2)):RANDOMIZE T
120 GOSUB 130:GOTO 150
130 LOCATE 22,1:LET FF$=STRING$(35,196)
140 BA$=FF$+STRING$(10,219)+FF$:PRINT BA$:RETURN
150 LOCATE 1,1
160 FOR J=1 TO 5
170 READ X:AB$="":GR1(J)=X:IF X=0 THEN GOTO 200
180 AB$=STRING$(80,X)
190 PRINT AB$:AB$:AB$:AB$:AB$:
200 NEXT J
210 FOR I=1 TO 6:READ PY(I),PX(I):NEXT I
220 RESTORE
230 REM-----Bucle general
240 LOCATE CY,CX:GOSUB 640:GOSUB 130
250 MEM$=INPUT$(1):JU$=" ":IF MEM$<>" " THEN GOTO 250
260 JU$=JU$+INKEY$:JU$=RIGHT$(JU$,1):IF JU$=" " AND GAS>0 AND CY>0 THEN LET VY=-1:GAS=GAS-10 ELSE IF CY=0 AND GAS>0 AND JU$=" " THEN VY=0:SOUND 751,1 ELSE VY=1
270 CY=CY+VY:JU$=""
280 IF CY=15 THEN GOSUB 640:GOTO 570
290 LET CX=FNVIEN(CY)+CX:CY1=INT(CX)
300 IF CX<1 THEN LET CX=68:CY1=68:GOTO 320
310 IF CX>68 THEN LET CX=2:CY1=2:GOTO 320 ELSE GOTO 340
320 SOUND 1046.8,.5:SOUND 987.7699,.8:SOUND 880,.8:LOCATE 1,1
330 FOR J=1 TO 4:AB$=STRING$(10,GR1(J)):AB$=AB$+STRING$(60,26)+AB$:PRINT AB$:AB$:AB$:AB$:NEXT J
340 R=INT((RND*150-59)/50):RR=INT(RND*6)+1:IF RR>3 THEN GOTO 360 ELSE I=RR
350 IF SGN(VV(I))<>SGN(R) THEN VV(I)=VV(I)+SGN(R)
360 IF VV(I)=0 THEN LET VV(I)=SGN(R)
370 GOSUB 640
380 REM-----PROYECTILES
390 NP=INT(RND*6)
400 IF PX(NP)=0 THEN PX(NP)=1:GOSUB 790:VP(NP)=4 ELSE IF PX(NP)=81 THEN PX(NP)=7
410 VP(NP)=-4:GOSUB 790
420 FOR I=1 TO 6:IF PX(I)=0 OR PX(I)=81 THEN GOTO 540
430 SS=FNCOL(PY(I)):SS$=CHR$(SS)
440 IF PX(I)>68 AND VP(I)>0 AND PX(I)<=80 THEN LOCATE PY(I),PX(I):PRINT SS$:PX(I)=81:GOTO 540
450 IF PX(I)>0 AND PX(I)<5 AND VP(I)<0 THEN LOCATE PY(I),PX(I):PRINT SS$:PX(I)=0:GOTO 540
460 IF PX(I)<1 OR PX(I)>80 THEN GOTO 500
470 IF VP(I)+PX(I)>80 THEN M=80 ELSE M=VP(I)+PX(I)
480 IF VP(I)+PX(I)<1 THEN M=1 ELSE M=VP(I)+PX(I)
490 FOR T=PX(I)+VP(I)/4 TO M STEP VP(I)/4:IF SCREEN(PY(I),T)<>SS THEN GOSUB 820:GOTO 500
500 NEXT T
510 IF BG=1 THEN GOTO 570
520 IF PX(I)=0 OR PX(I)=81 THEN GOTO 540
530 IF VP(I)>0 THEN D=5:DD=29 ELSE D=3:DD=28
540 PX(I)=PX(I)+VP(I):IF BG<>1 AND PX(I)>0 AND PX(I)<81 THEN LOCATE PY(I),PX(I):PP$=CP$(I)+STRING$(D,DD)+SS$:PRINT PP$:
550 NEXT I
560 LOCATE 24,20:PRINT "SCORE ":SCO:LOCATE 24,50:PRINT "GAS ":GAS:
570 IF GAS<100 THEN PLAY "MB AB":LOCATE 24,60:PRINT "¡PELIGRO!";
580 GOTO 260
590 LOCATE 1,20:IF CX>31 AND CX<40 AND BG<>1 THEN PRINT " FELICITACIONES:¡ATERRI-
ZAJE CORRECTO.¿OTRA PARTIDA?":PLAY "O1DFDBDEFG00B" ELSE LOCATE 1,30:PRINT "PERDI-
STE.¿OTRA PARTIDA?":PLAY "00BAGF"
600 LOCATE 2,25:PRINT "(TECLEA RESPUESTA Y PULSA RETURN)";
610 INPUT Q$:Q$=LEFT$(Q$,1):IF Q$="S" OR Q$="s" THEN GOTO 50 ELSE CLS:KEY ON:END

```



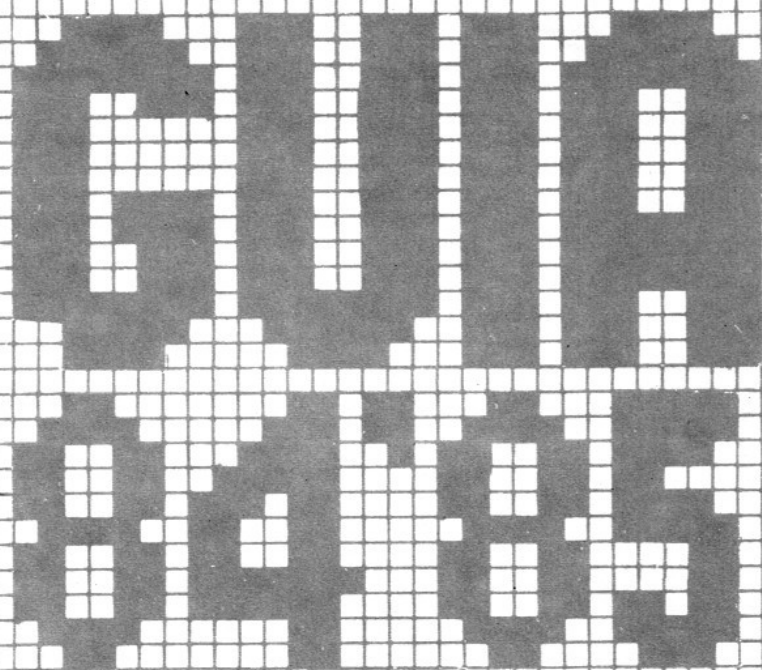
```

600 REM-----DATOS
610 DATA 178,177,176,000,000
620 DATA 3,0,5,81,8,0,10,81,13,0,15,81
630 REM-----DIBUJO DEL GLOBO
640 LOCATE 1,1:CC$=CHR$(31)+STRING$(10,29):GB$=STRING$(CX1,28)+STRING$(CY,31)
650 IF VY>0 THEN C1=FNCOL(CY+1):W$=STRING$(10,C1):GB$=GB$+W$+CC$ ELSE GB$=GB$+CH
R$(31)
660 C1=FNCOL(CY+2):W$=CHR$(C1)+CHR$(C1):GB$=GB$+W$+" "W$+CC$
670 C1=FNCOL(CY+3):C$=CHR$(C1):W$=STRING$(8,""):GB$=GB$+C$+W$+C$+CC$
680 C1=FNCOL(CY+4):C$=CHR$(C1):W$=STRING$(6,""):GB$=GB$+C$+" "W$+" "C$+CC$
690 C1=FNCOL(CY+5):W$=STRING$(4,C1):GB$=GB$+W$+" "CHR$(C1)+W$+CC$
700 C1=FNCOL(CY+6):W$=STRING$(4,C1):GB$=GB$+W$+" "W$+CC$
710 IF VY<0 THEN C1=FNCOL(CY+7):W$=STRING$(10,C1):GB$=GB$+W$
720 PRINT GB$;:GOSUB 130
730 VI$=STRING$(2,31)
740 FOR I=1 TO 3:LOCATE (I-1)*5+3,1:ON SGN(VV(I))+2 GOTO 750,760,770
750 PRINT "<";:GOTO 780
760 PRINT " ";:GOTO 780
770 PRINT ">";
780 NEXT I:RETURN
790 CC=INT(RND*5):ON CC GOTO 800,800,810,810,810
800 CP$(NF)=" ":RETURN
810 CP$(NF)=" ":RETURN
820 REM-----IMPACTOS Y PUNTOS
830 IF SCO=1500 OR SCO=3000 THEN GAS=GAS+300:PLAY "DDD"
840 IF CP$(I)=" " THEN SCO=SCO+100:SOUND 250,1:SOUND 680,1:LOCATE FY(I),PX(I):PR
INT SS$:IF VP(I)>0 THEN PX(I)=0:RETURN ELSE PX(I)=81:RETURN
850 IF CP$(I)<>" " THEN RETURN
860 LOCATE FY(I),PX(I)-4:PRINT "!!!BANG!!!!":BG=1
870 FOR Z=1 TO 360 STEP 18:SE=500+SIN(Z)*200:SOUND SE,.5:NEXT Z:RETURN

```



**DENTRO
DE POCO
APARECERA
LA**



**RESERVELA
EN SU KIOSKO HABITUAL
EN TODAS LAS BUENAS TIENDAS DE INFORMATICA
O PIDALA A: EL ORDENADOR INDIVIDUAL, S.A.
FERRAZ, 11-3º- 28008-MADRID**

MPF·V~PC/XT

PC COMPATIBLE con su economía.

CONFIGURACION BASICA

UNIDAD CENTRAL

- Microprocesador 8088, de 16 bits, 477 MHz.
- 8 K Bytes ROM (Ampliables a 48KB)
- 256 K Bytes RAM

DISPLAY

- 8 páginas de texto, de 40 x 25 caracteres, 4 páginas de 80 x 25,8 colores en borde y 16 para la pantalla
- Gráficos de 320 x 200, con 4 colores, 640 x 200, en blanco y negro

SALIDAS VIDEO

- Video compuesto
- TTL - RGB
- TV

TECLADO

- Ergonómico 90 teclas
12 de función
4 definibles por usuario numérico independiente

UNIDAD DE DISCO

MPF-V.PC

- 2 Driver Floppy Disk con 360 KB cada uno

MPF-V.XT

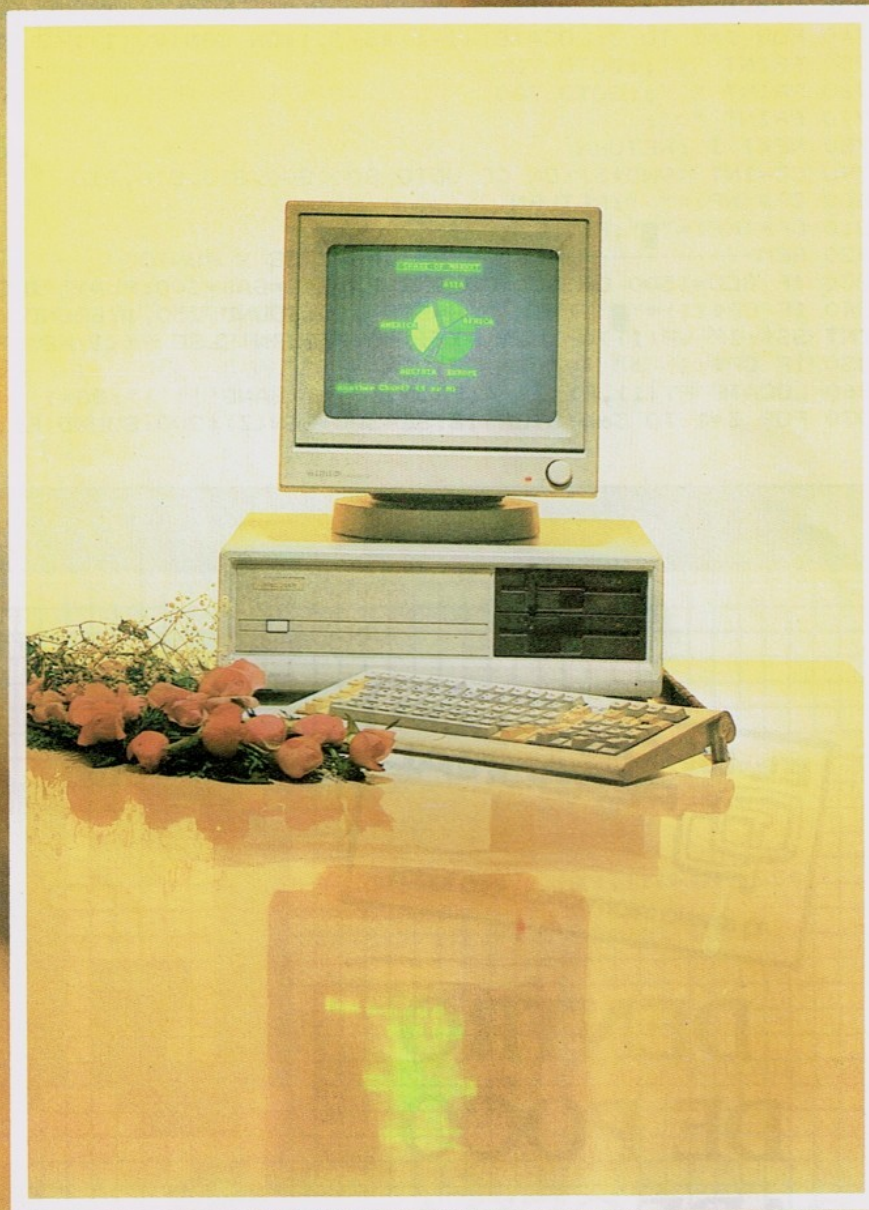
- 1 Driver Floppy Disk con 360 KB
- 1 Disco Winchester 10 MB

DISPOSITIVO DE SALIDA

- Interface de RS232C
- Interface paralelo Centronics
- 5 slots de salida compatible con IBM PC™

SISTEMAS OPERATIVOS

- MS-DOS™ V2.11 (Incluido)
- CC P/M-86™ Concurrente Multitarea, ventanas de video PC-Mode (Incluido)
- PC-DOS™ (soportado)



MS DOS 2.0 es marca registrada de Microsoft Inc. CCP/M 86 es marca registrada de Digital Research Inc.
PC-DOS e IBM PC son marcas registradas de International Business Machines Corporation.



CECOMSA

Castelló, 25, 3.º E - Madrid-1 - Teléf. 435 37 01



Enseñe al ordenador geografía de España, y él se encargará de que sus hijos la aprendan jugando

El aprendizaje tiene dos fases. Primero debe adquirir los conocimientos la máquina (por supuesto a partir de lo que el hombre sabe). Una vez que los asimila, puede recordarlos para siempre, y convertirse a su vez, en maestro en el tema (limitadísimo) para el que ha sido adoctrinada. Tomando como base la división en provincias de España, se ha estructurado un programa capaz de aprender y utilizar sus conocimientos.

Introducción

La estructura de la información en forma de árbol (Figura 1^a) fué ya tratada en el número 1. El almacenamiento de las preguntas y respuestas sobre geografía va a realizarse sobre un árbol, que en el programa BASIC se almacena en una tabla (figura 1 b),

y en la memoria del ordenador como una lista. (Figura 1 c).

El árbol tratado es un árbol dicotómico, en que, en cada nudo, pueden tomarse dos y sólo dos alternativas. El árbol termina en "hojas" y éstas se caracterizan por ser la información final y tener como valor de los apuntadores SI y NO, cero.

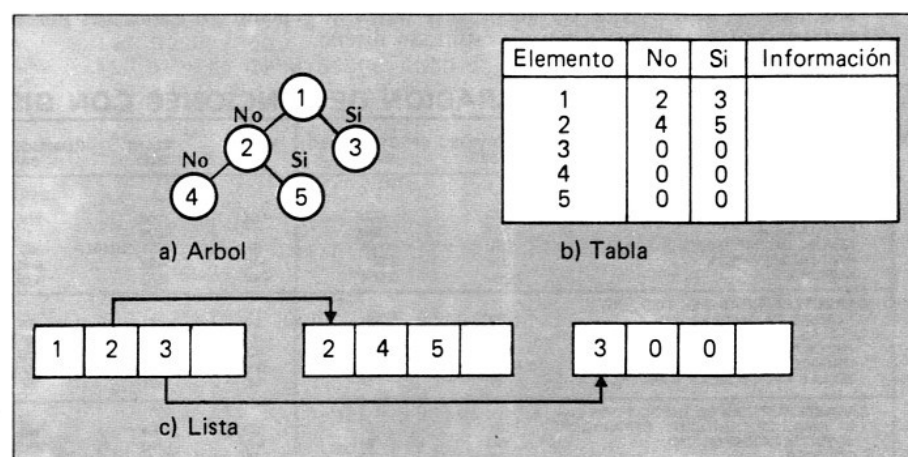


Figura 1. Árbol, Tabla, Lista.

QUIERE RECURRIR A CONOCIMIENTOS PREVIOS (SI ó NO)? SI
 ESTA EN LA PENINSULA (SI ó NO)? NO
 ES UNA ISLA (SI ó NO)? SI
 SE TRATA DE MALLORCA, ESTA DE ACUERDO (SI ó NO)? SI
 QUIERE SEGUIR (SI ó NO)? NO
 QUIERE SALVAR LA INFORMACION EN DISCO (SI ó NO)? NO

Figura 2. Diálogo.

SPECTRAVIDEO

El sistema con el que usted crecerá



Los ordenadores personales de Spectravideo, gracias a su perfecto diseño y gran facilidad de manejo, se pondrán enteramente en sus manos desde el momento que los desembale. Y, aún más, son suficientemente capaces y ampliables como para cubrir todas las necesidades durante mucho tiempo.



SV-318 SV-328

- **BASIC extendido de Microsoft**, integrado, convierte a Spectravideo en los primeros ordenadores verdaderamente programables y asequibles del mercado.
- **Extraordinaria memoria**. 32 Kbytes de ROM, ampliables a 96 Kbytes; y 32 ó 80 Kbytes de RAM, ampliables a 256 Kbytes.
- **Expansión diagonal**. Un sistema de 14 periféricos plenamente soportado. Incluyendo el adaptador de juegos Colecovisión. Unidad de expansión de 7 slots, controlador de disco flexible, cassette, interface para cartucho, etc.
- **Gráficos avanzados**. El sistema SV ofrece 16 colores en alta resolución y, más importante aún, 32 sprites programables que permiten unas impresionantes posibilidades de control de animación en pantalla.
- **Otras muchas características atractivas**. Tales como un microprocesador Z80A con un rápido (3,6) reloj interno, slot para la conexión de cartuchos, diez teclas de función programables por el usuario, tres canales de sonido (8 octavas por canal), perfil bajo y estilizado diseño.

TABLA DE COMPARACION DE FUNCIONES CON DISTINTAS MARCAS

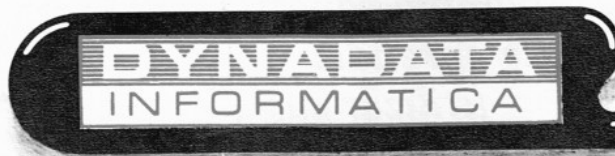
	SPECTRAVIDEO SV-328	SPECTRAVIDEO SV-318	APPLE IIe	ATARI 800	COMMODORE 64	BBC MODEL B	DRAGON 32	EL MAS VENDIDO HASTA AHORA
CAPACIDAD DEL ORDENADOR								
ROM INCORPORADA	32k	32K	16K	10K	20K	16K	16K	16K
AMPLIABLE A	96k	96K	?	42K	?	64K	?	?
MBASIC INCORPORADO	SI	SI	SI	ADICIONAL	NO	NO	SI	NO
RAM INCORPORADA	80k*	32K*..	64K	48K	64K	32K	32K	16K
AMPLIABLE A	256k**	256K*..	64K	NO	N/A	32K	64K	48K
CARACTERISTICAS DEL TECLADO								
NUMERO DE TECLAS	87	71	63	61	66	73	53	40
TECLAS DEFINIBLES POR EL USUARIO	10	10	?	4	8	10	?	?
PROCESO DE TEXTOS	SI	SI	NO	NO	NO	NO	NO	NO
GRAFICOS (DESDE TECLADO)	SI	SI	NO	SI	SI	SI	SI	SI
TECLAS MAYUSCULAS Y MINUSCULAS	SI	SI	SI	SI	SI	SI	SI	SI
CARACTERISTICAS DE JUEGO Y SONIDO								
RANURAS DE CARTUCHOS SEPARADAS	SI	SI	NO	SI	NO	NO	SI	NO
JOYSTICK INCORPORADO	NO	SI	NO	NO	NO	NO	NO	NO
COLORES	16	16	15	128	16	16	9	8
RESOLUCION (PIXELS)	256x192	256x192	280x180	320x192	320x200	256x640	256x192	256x192
SPRITES	32	32	N/A	4	8	?	16	?
CANALES DE SONIDO	3	3	1	4	3	1	3	?
OCTAVAS POR CANAL	8	8	4	4	9	3	5	3
ENVOLUTURA A.D.S.R.	SI	SI	NO	NO	SI	SI	NO	NO
CAPACIDAD DE LOS DISCOS	256K	256K	143K	92K	170K	100K	?	?
COMPATIBILIDAD CON CP/M (programas standard de 80 columnas)								
CP/M 2.2	SI	SI	NO****	NO	NO****	SI	NO	NO
CP/M 3.0	SI	SI	NO	NO	NO	NO	NO	NO
MSX	SI	SI	NO	NO	NO	NO	NO	NO

Microsoft es una marca registrada de Microsoft Corporation
CP/M, es una marca registrada de Digital Research, Inc.

Las especificaciones están sujetas a cambios sin previo aviso.
Comparación efectuada en febrero 84

- ... 64 K disponibles con CP/M 2.2, más 16 K de soporte gráfico
- ... 240 K disponibles con CP/M 3.0, más 16 K de soporte gráfico
- ... 16 K disponibles, más 16 K de soporte gráfico
- ... Apple II acepta con una tarjeta de modificación de 40 ó 80 columnas CPM
- ... Comodore 64 acepta 40 columnas CPM
- ? Dato no confirmado

- Garantizamos el mantenimiento de nuestros equipos.



Planteamiento

El programa gestiona el árbol en dos modalidades. La primera recorriéndolo y mostrando la información contenida en él. La segunda introduciendo nueva información, con lo que ésta operación conlleva. ¿Y qué conlleva? Pues el cambio de los apuntadores SI y NO de la hoja (elemento terminal) que se va a matizar con una nueva pregunta.

Parte de un árbol mínimo con

1	2	3	ESTA EN LA PENINSULA?
2	0	0	CEUTA
3	0	0	ZARAGOZA

para evitar los problemas de creación inicial del árbol. De esta manera el programa tiene en la tabla elementos para hacer la primera pregunta y dar la primera respuesta. Si la respuesta es válida todo bueno. Si no lo es puede declarar.

"Me rindo, si no es A es que piensa".

Aquí empieza el proceso de aprendizaje. Le decimos la provincia que pensábamos es B, y nos pide un elemento de diferenciador entre las dos provincias. Este concepto diferenciador, será la pregunta que realice en la siguiente utilización. Como el elemento diferenciador entre A y B puede no quedar claro a quien se aplica, nos pregunta si A tiene el elemento diferenciador, para de no ser así, aplicarlo a B (lo que se traduce en la generación de los enlaces adecuados).

Enriquecido el sistema con un nuevo dato, o acertada por el ordenador, la provincia que pensábamos, nos pregunta si deseamos continuar en el juego. De ser SI, se comienza de nuevo la adivinanza de una provincia. De ser NO, se nos pregunta si queremos guardar la "sabiduría" en diskette, para utilizarla en una ejecución posterior. En cualquier caso,

ELEM	NO	SI	CONTENIDO
1	2	3	ESTA EN LA PENINSULA?
2	14	15	ES UNA ISLA?
3	4	5	ESTA EN EL MAR?
4	12	13	ES LA CAPITAL DE ESPAÑA?
5	6	7	ESTA MUY AL SUR?
6	9	8	TIENE MAR MENOR?
7	0	0	CADIZ
8	0	0	MURCIA
9	10	11	TIENE FALLAS EN SAN JOSE?
10	0	0	CARTAGENA
11	0	0	VALENCIA
12	0	0	ZARAGOZA
13	0	0	MADRID
14	0	0	CEUTA
15	0	0	MALLORCA

Figura 3. Volcado de la tabla.

```

10 REM PROGRAMA GESTOR DE ARBOLES DICOTOMICOS
15 REM APLICADOS AL ESTUDIO DE LA GEOGRAFIA
20 REM EL ORDENADOR PERSONAL
25 REM
30 REM POR MIGUEL SOLANO
40 REM 06:03:92
50 REM
90 R$=CHR$(13)
100 GOSUB1000
110 PRINT"QUIERE RECURRIR A CONOCIMIENTOS PREVIOS ";
111 GOSUB9800
120 IF I=1 THEN GOSUB 2000
130 P=1:R=0
135 PRINT"J"
140 PRINTT$(P);:GOSUB9800
150 R=E(P,I)
160 IF E(R,0)=0GOTO200
170 P=R
180 GOTO140
200 PRINT"SE TRATA DE ";T$(R);", ESTA DE ACUERDO ";
201 GOSUB9800
220 IF I=1 GOTO 405
250 PRINT"ME RINDO, SI NO ES ";T$(R);", EN QUE PIENSA ";
260 INPUT N$
270 PRINT"QUE DIFERENCIA A ";T$(R);" DE ";N$;" ";
280 INPUT D$
290 PRINTT$(R);" ";D$;:GOSUB9800
350 T$(L)=T$(R)
355 T$(L+1)=N$
360 E(R,0)=L+I
365 E(R,1)=L-I+1
370 T$(R)=D$
380 L=L+2
400 PRINT"GRACIAS POR UN NUEVO DATO, ";
405 PRINT"QUIERE SEGUIR ";:GOSUB9800
410 IF I=1 THEN GOTO130
450 PRINT"QUIERE SALVAR LA INFORMACION EN DISCO ";
451 GOSUB9800
460 IF I=0 THEN PRINT:GOSUB3000:END
470 GOSUB2500
480 GOSUB3000
990 END
991 REM
992 REM
993 REM
994 REM
1000 REM Rutina de INICIALIZACION
1001 REM -----
1010 DIM E(100,1),T$(100)

```


DRAGON Data Ltd.



Extensa variedad de software comercial: BASE DE DATOS • PROCESADOR DE TEXTOS • CONTABILIDAD
CONTROL DE ALMACENES • ETIQUETAS
GENERADOR DE NOMINAS • TIENDAS DE CALZADO • FACTURACION • VIDEO CLUB • CURSO COMPLETO DE BASIC, ETC., ETC., ETC.

... Y los mejores juegos existentes en el mercado mundial:
AJEDREZ - BATALLA NAVAL - EL AHORCADO - SIMULADOR DE VUELO - ATTACK - CAVE HUNTER - ETC., ETC., (HASTA 300 JUEGOS DIFERENTES)

¡¡Todos disponibles en CASSETTE y en DISQUETE de 5 1/4"!!

Solicita, sin compromiso, relación de software, libre de todo gasto.

DE VENTA EN DISTRIBUIDORES AUTORIZADOS

IDS

Informática y desarrollo de Sistemas, S.A.



IMPORTADOR EXCLUSIVO

Español eto, 25

Madrid-4

Telf.: 410 30 64

GRUPO

CODERE BARCELONA, S. A.

BERLIN, 50 - 52

230 61 05 - 239 50 06

BARCELONA-29

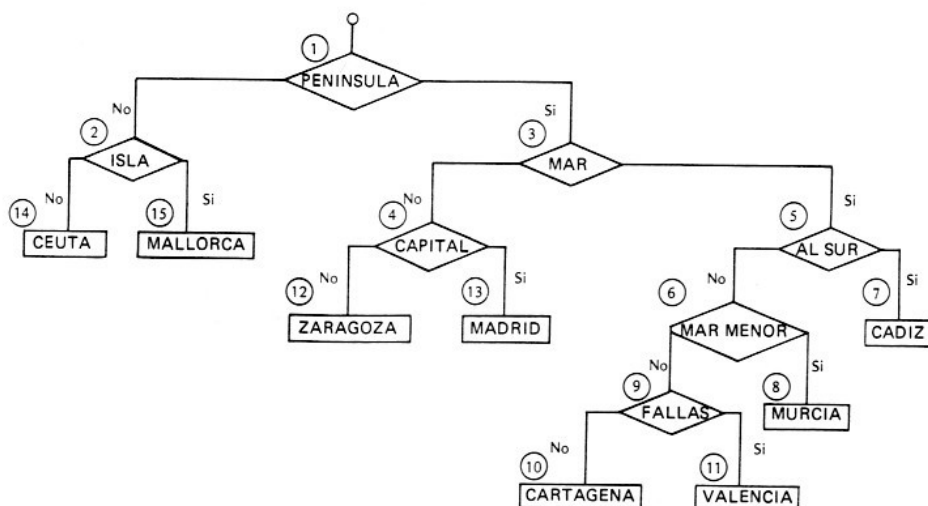


Figura 4. Árbol dicotómico correspondiente a la tabla.

```

1020 L=4
1030 E(1,0)=2:E(1,1)=3:T$(1)="ESTA EN LA PENINSULA"
1040 E(2,0)=0:E(2,1)=0:T$(2)="CEUTA"
1050 E(3,0)=0:E(3,1)=0:T$(3)="ZARAGOZA"
1100 PRINT"J"
1990 RETURN
1999 REM
2000 REM CARGA DE DISCO
2001 REM -----
2010 DOPEN#130,"GEOGRA",DO
2050 INPUT#130,L
2060 FOR K=1 TO L
2070 INPUT#130,E(K,0),E(K,1),T$(K)
2080 NEXT
2090 DCLOSE#130
2490 RETURN
2498 REM
2499 REM
2500 REM SALVADO A DISCO
2501 REM -----
2510 OPEN2,8,2,"@:GEOGRA,S,W"
2520 PRINT#2,L;R$;
2530 FOR K=1 TO L
2540 PRINT#2,E(K,0);R$:E(K,1);R$:T$(K);R$;
2550 NEXT K
2560 CLOSE 2
2990 RETURN
2999 REM
3000 REM RUTINA DE LISTADO DE LA TABLA
3001 REM -----
3002 OPEN4,4:CMD4
3003 PRINT"ELEM","NO","SI","CONTENIDO"
3004 PRINT
3005 FOR K=1 TO L-1
3010 PRINT K,E(K,0),E(K,1),T$(K);
3012 IF E(K,0)=0 THEN PRINT:GOTO 3020
3014 PRINT" ?"
3020 NEXT K
3030 PRINT"YA LA HA VISTO ";:GOSUB9800
3040 RETURN
3099 REM
9800 REM RUTINA DE SI O NO
9801 REM -----
9810 INPUT" (SI O NO) ";S$
9820 S$=LEFT$(S$,1)
9830 IF S$="N" THEN I=0:RETURN
9840 IF S$="S" THEN I=1:RETURN
9845 IF S$="C" THEN I=2:RETURN
9850 PRINTCHR$(7);:GOTO9810
9997 REM

```



imprime la tabla de enlaces E (L,2) y la de preguntas y respuestas T\$(L) siendo L el número de elementos.

Al iniciar el juego, podemos hacerlo recurriendo a lo aprendido en el juego anterior, o bien con la memoria en blanco.

Programa.

El programa principal está contenido en las líneas 10 á 990 y hace uso de subrutinas.

La subrutina que comienza en 1000 inicializa el árbol y asigna valores.

En 2000 se toman los datos de diskette (secuencial).

En 2500 se salvan los datos a diskette.

En 3000 se listan las tablas.

En 9800 se codifica la rutina que siempre pregunta y válida por el SI y el NO.

Ampliaciones.

Sobre el esquema que ha presentado y fiel a mi línea de dar ideas sobre las que desarrollar programas más complejos, realizo las siguientes sugerencias.

a) Al llegar a un pregunta dada, que se nos proponga SI, NO ó MATIZAR. Si se elige la 3ª vía, se creará un nuevo nudo, insertándolo antes de la pregunta recién formulada, que permitirá un retoño adicional. Han de gestionarse los enlaces.

b) Admitir árboles n-tónicos, es decir con n posibilidades o retoños en cada nudo. Este problema es algo complicado y si en dos meses no está realizado, propondré la solución.

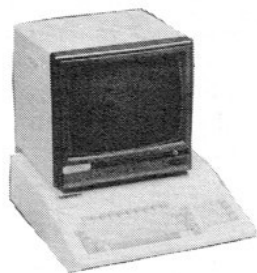
Nota.

El programa ha sido ejecutado sobre un CMB Modelo 8032 de Commodore.

Miguel Solano.

Un modelo para cada necesidad Un precio para cada economía

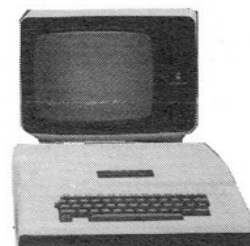
Compatibles Apple



Microordenador APOLLO
Mod. Ami 64 K doble CPU
(6502 + Z80). Compatible
Apple 118.000 Ptas.



Microordenador GOLDEN II.
48 K. Memoria ampliable.
Z. 80 opcional. Compatible
Apple 89.000 Ptas.



Microordenador SPRING 48 K
Memoria ampliable, Z-80
opcional. 80.000 Ptas.



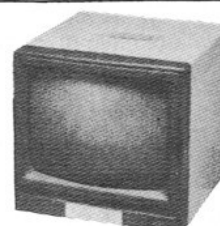
16 Bit compatible IBM.
CPU 808 B-2 DRIVES-carta para .co-
lor-Monitor B/N color 128 KRAM ex-
pandible a 256 K.



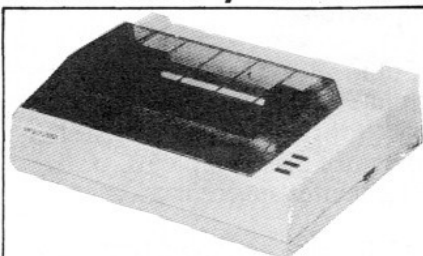
AMI 304
Compatible Apple con 256 KRAM-
microprocesador 8 bit con bus de
16-80 columnas incluido.



Multiprogramador de
EPRONS: LEP. 1200 para grabar 10
EPRONS a la vez para las 2716, 2732,
2732A, 2764A, 2516, 2532, 2564,
68766, 27128, o cualquier otra que
sea compatible.



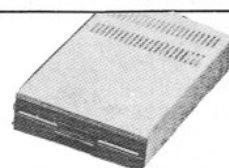
Monitores 12" y 9" verde o
naranja 29.600 Ptas.



Impresora CIEN-80 C
82.000 Ptas.



Fichero 50 discos 5 1/4"
3.500 Ptas.



Phloppies extraplanos
diversas capacidades
Tarjetas opcionales
accesorios

Manuales y sistemas operativos en español

APPLE-COMMODORE-SINCLAIR-COMPATIBLES AMI-256

Y muchos complementos más

IEESA MICROTHERSA

C/ Miguel Yuste, 16. Tel. 204 51 98.
Telex: 49249 / MADRID-28037.

Se sirve a provincias. Distribuidores.
Consulten sus precios.

¡ES DIFÍCIL, PERO SI NO LO TENEMOS LO
BUSCAMOS PARA UD.
¡SIN RECARGO! ¡Y ADEMÁS SERVIDO DESDE
MADRID!!

BOLETIN DE PEDIDO

Nombre 1º Apellido

2º Apellido

Calle Piso Ciudad Prov.

- ☐ Adjunto talón conformado núm.
☐ Con cargo a mi tarjeta visa núm.
☐ Envío contra reembolso con gastos a mi cargo
☐ Solo información.

Banco
Ptas.

CANTIDAD	CONCEPTO	PRECIO/U	TOTAL

Pánico en el fondo del mar

Se encuentra Vd. al mando de un submarino que posee un tubo lanzatorpedos. ¿Su fin? Abatir la mayor cantidad posible de adversarios. Tranquilícese: a falta de espacio intergaláctico, atravesará las marejadas de la alta resolución del VIC 20.

Los submarinos enemigos nos rodeaban por todas partes, habían sido guiados por un satánico helicóptero que nos había causado graves daños. Me disponía a lanzar los torpedos cuando un pulpo gigante con enormes tentáculos...



Situado en el fondo del mar, está Vd. en posesión de un arma que le permitirá destruir tres tipos de enemigos: un helicóptero (sobrevolando el mar, por supuesto) lo que le supondrá 100 puntos, dos barcos por los que se beneficiará de 70 puntos, y un submarino gracias al cual ganará 200 puntos.

En cada pasada el submarino deposita una mina con el único fin de hacerle perder 15 puntos, pero Vd. obtiene en tal caso un torpedo de más (dispone inicialmente de sesenta y seis torpedos). Si le acierta al submarino todas las minas presentes en la pantalla desaparecen.

Para evitar el tiro en ráfagas, un test verifica que Vd. no haya tirado tres veces sucesivas desde el mismo sitio, en cuyo caso el cañón permanecerá inactivo (tendrá que desplazarse para poder utilizarlo normalmente).

El tiro del último torpedo provoca la aparición en pantalla de su puntuación definitiva, así como el fin del juego si no ha podido sobrepasar los 1.500 puntos necesarios para jugar una vez más.

*Un programa con o sin
mando de juegos*

El programa, tal y como está escrito, necesita el empleo de un mando de juegos (CBM o Atari); pero la tabla 1

READY.

```

1 POKE36879,29:PRINT"XXXXXXXXXXXXFACIL= /F/"
2 PRINT"XXXXXXXXXXXXFACIL= /D/"
3 P=PEEK(197):IFP=18 THEN 00:GOTO9
5 IFP=42THEN00=1:GOTO9
7 GOTO3
9 PRINT"D"
10 POKE36869,255:POKE52,28:POKE56,28
11 IF 00=1 THEN CLR:00=0
12 IF 00=0 THEN CLR:00=0
13 POKE36878,10
20 A=7168:FORI=ATORA+511:POKEI,PEEK(I+25600):NEXT
30 FORL=0TO111:READF:POKEA+L,F:NEXT
40 DATA1,0,0,32,119,32,0,0
45 DATA0,3,7,15,255,114,63,31
50 DATA3,3,3,127,255,213,255,127
55 DATA255,16,56,100,230,124,40,124
60 DATA0,192,224,240,255,78,252,248
65 DATA128,128,128,254,255,171,255,254
70 DATA16,16,16,56,254,254,0,16,16,0,0,16,16,0
75 DATA0,82,62,252,63,122,54,16,128,81,42,4,0,0,0,0,0
77 DATA1,130,64,28,62,28,0,0,16,16,16,56,56,0
79 DATA0,0,3,127,7,63,21,31,0,8,144,160,192,252,88,240
80 U=37154:V=37137:W=37152:Z=8189:S=7834:0=0:I=8186
85 FORR=38818TO38839:POKER,2:NEXT
87 FORR=8128TO8186:POKER,11:POKE30720+R,3:NEXT
90 FORR=7855TO7877:POKER,9:POKE30720+R,6:NEXT
100 POKEU,127:V=-(PEEK(W)AND128)=0:X=-(PEEK(V)AND16)=0:POKEU,255
110 D=1:Z=Z+X+Y:IFZ<80990RZ=8118THEN200
115 IFZ1<ZTHEN0=0:POKEZ1,32
120 POKEZ,6
122 Z1=Z:T=PEEK(V)AND32
125 IFQ<2THEN200
130 IFQ<32THEN:D=2:Q=Q+1:GOTO600
200 OND1GOTO250,300,400
250 POKES,32:S=S+M:POKE30720,4
260 IFS=7834ORS=7855THENPOKES,32:GOTO500
280 POKES,1+F1:POKE30720,4
290 D1=1:GOTO450
300 POKES,32:S=S+M:POKE30720,2
310 IFS=7878ORS=7899THENPOKES,32:GOTO500
320 POKES,2+F1:POKE30720,2
330 IFS=CTHENPOKEC-22,10
360 D1=2:GOTO450
400 POKE36877,200:POKE3,32:POKE3+1,32:S=S+1
410 IFS=7833THENPOKE36877,0:GOTO500

```

```

420 POKES,0:POKE3+1,3:D1=3:POKE36877,0:POKE3+30721,7
450 ONDGO100,610
500 N=NRND(1):M=INT(NRD(1)+.5)
510 IFM=0THENM=-1:F1=3:F2=0:GOTO530
520 F1=0:F2=3
525 IFN<25THEND1=1:S=7854:M=-1:F1=12:F2=11:GOTO450
530 IFN<5THEND1=1:S=7845-9*M:GOTO450
540 IFN<75THEND1=2:S=7889-9*M:C=Z-220:GOTO450
550 D1=3:S=7812:M=1:GOTO450
600 R=Z:D=2:IFII<8186THENII=8186
625 II=II+1:IFII<8120THEN1500
607 POKEII,32
610 FORR=0TO000:POKE36877,220:IFR<7900THEN650
620 R=R-22
630 POKE7,7:POKE3+30720,7:POKE36877,0:D=2
640 NEXT:GOTO200
650 SI=R-S:SE=R-S-M
655 IFPEEK(Z-242)=10THENS=SC-15:POKEII,11:II=II+1:POKEZ-242,8:GOSUB845
660 IFZ2<INT(81/22)=SITHENSA=PEEK(S):GOSUB900:GOTO680
670 IFZ2<INT(81/22)=SITHENSA=PEEK(S+M):GOSUB900
680 FORR=Z-22TOZ-220STEP-22
690 POKE7,32:NEXT:POKE36877,0:GOTO100
800 IFPEEK(Z-242)=10AND(S=20RSA=5)THENS=SC+80
810 IFS=20RSA=5THENS=SC+300:GOSUB900:GOTO840
815 IFS=10RSA=40RSA=120RSA=13THENS=SC+70:GOTO840
820 IFS=00RSA=3THENS=SC+100
840 POKES,0:POKE3+30720,7:POKE36878,15
845 FORL=0TO5:POKE36875,220:GOSUB900:POKE36875,0:GOSUB900:NEXT
850 POKEZ-242,9:POKE36878,10:GOTO1000
900 FORA=1TO30:NEXT:RETURN
950 IFD1=2THENFORR=7856TO7877:POKER,9:NEXT
960 RETURN
1000 FORA=7717TO7723:POKER,32:NEXT
1010 SC=STR$(SC):L=LEN(SC)
1020 FORA=1TOL
1030 POKE7717+A,ASC(MID$(SC$,A,1))+128
1040 POKE38438+A,0:NEXT:RETURN
1500 PRINT"D":POKE36869,240
1510 PRINT"XXXXXXXXXXXXMUNICIONES="SC
1520 IFSC<1500THENPRINT"XXXXXXXXXXXXMUNICIONES="SC
1530 PRINT"XXXXXXXXXXXXMUNICIONES="SC
1560 GET A$
1570 IFA$<"0"THEN RUN
1575 IFA$<"N"THENGOTO 1560
1580 END

```

READY.

PROGRAMACION DE ORDENADORES EN BASIC



un nuevo libro de la colección PROCESO DE DATOS

POR JESUS SANCHEZ IZQUIERDO Y FRANCISCO ESCRIBUELA VERCHER

- UN LIBRO QUE ENSEÑA LOS CONOCIMIENTOS DE UNO DE LOS LENGUAJES MAS SIMPLS Y A LA VEZ MAS EFICACES DE PROGRAMACION: EL BASIC
- UN LIBRO EMINENTEMENTE PRACTICO EN QUE CADA PASO QUEDA MATIZADO POR UN GRAN NUMERO DE EJEMPLOS RESUELTOS.
- UN LIBRO COMPLETO, REDACTADO EN FORMA CLARA Y CONCISA.
- UN LIBRO ABSOLUTAMENTE NECESARIO PARA TODOS LOS USUARIOS DE ORDENADORES QUE REQUIERAN DE ESTE TIPO DE LENGUAJES CONVERSACIONALES.
- SIN DUDA, EL LIBRO QUE ESPERABAN LOS USUARIOS PRESENTES Y POTENCIALES DEL BASIC.

HAGA SU PEDIDO A PROCESO DE DATOS. FERRAZ 11 - MADRID - 8. Precio 940-PTAS

Deseo recibir ejemplares

Sr.
 Empresa
 Cargo
 Domicilio
 Población
 Provincia

Forma de pago:

- ☐ Talón adjunto a nombre de Prodaee, S.A.
☐ Giro postal nº Fecha ...
☐ contra reembolso.

le permitirá utilizar en su lugar el teclado sin, por ello, alterar la esencia del juego.

Por otra parte, y con el fin de no sobrepasar la capacidad de memoria del sistema (3583 octetos libres), no se ha incluido ningún comentario en el programa. (Puede consultar la tabla 1 en la que se incluye una breve descripción del mismo).

Tal vez esté Vd. sorprendido del imponente tamaño del programa para un juego de esta naturaleza. Podría, en efecto, ser más "económico" insertar una rutina de visualización y desplazamiento de caracteres en un bucle FOR ... NEXT y utilizarla sucesivamente para cada adversario (cuatro enemigos, el cañón, la ráfaga). Pero este método presenta la gran desventaja de necesitar una batería impresionante de preguntas del tipo IF ... AND ... THEN o IF ... OR ... THEN, lo que se traduciría en una menor velocidad de ejecución. Por ello se ha escrito una rutina propia a cada objeto.

Si desea modificar los gráficos, puede recalcular los argumentos de los datos (líneas 40 a 79), siguiendo las indicaciones de la tabla 2.

El microprocesador 6502 "entiende" un carácter como una sucesión de ocho números X ($0 < X < 255$) siendo cada número X la representación decimal de un octeto. He aquí el procedimiento a seguir para obtener un nuevo carácter:

- Trazar, en una hoja de papel, una matriz de 8×8 casillas.
- Desarrollar en esta matriz el dibujo que queramos, teniendo en cuenta que una casilla sólo puede estar vacía o llena (0 ó 1).
- Por cada línea, hacer la suma de las ocho casillas que la componen. Una casilla i tiene valor 0 si está vacía, o el valor 2^i si está llena ($0 \leq i \leq 7$). La línea 0 es la de arriba, la columna 2^0 es la que está más a la derecha.

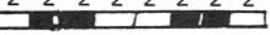
Tabla 1: Pequeños consejos para modificar el programa de forma que pueda utilizarse el teclado para controlar los desplazamientos y el tiro.	
Línea	Comentarios
80	quitar U = 37154 : V = 37137 : W = 37152 ; únicamente
100	reemplazarla por P = PEEK (197) : IF P = 31 THEN Z = Z - 1 : GOTO 110
105	introducir IF P = 23 THEN Z = Z + 1
110	quitar solamente Z = Z + X + Y :
122	quitar solamente : T = PEEK (V) AND 32
130	reemplazar T < > 32 por P = 8

Teclas a utilizar		
Tecla	Comentario	Código
↑ CRSR	desplazamiento a la izquierda	31
← CRSR →	desplazamiento a la derecha	23
←	disparo	8

Si desea cambiar de teclas, es suficiente reemplazar los códigos 31, 23 y 8 de las líneas 100, 105 y 130 por los de las nuevas teclas que Vd. elija.

Tabla 2: Cálculo de argumentos de los DATA para la modificación de gráficos		
Línea	Carácter del teclado o POKE ----, X	Comentarios
40	; X = 0	traseira (parte) del helicóptero)
45	A ; X = 1	popa del barco nº 1
50	B ; X = 2	popa del submarino
55	C ; X = 3	delantera del helicóptero
60	D ; X = 4	proa del barco nº 1
65	E ; X = 5	proa del submarino
70	F, G ; X = 6, X = 7	cañón, ráfaga
75	H, I ; X = 8, X = 9	explosión, ola
77	J ; X = 10	mina
79	K, L ; X = 11, X = 12	popa y proa del barco nº 2

- Llevar los ocho argumentos a su correspondiente DATA, por ejemplo:

Línea i $2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
 = 1
 octeto.

El valor de esta línea será: $Li = 0 + 64 + 32 + 0 + 0 + 4 + 2 + 0 = 102$

Vd. juega ahora. Si dispone de una extensión de memoria puede añadir todo tipo de florituras, como por ejemplo, gráficos más realistas o adversarios más correosos (pulpos gigantes o ferozes cetáceos). □

Patrick Lenoble



GESTION DE FICHEROS

Mantenimiento de una biblioteca

Introducción:

El presente programa realiza las operaciones oportunas para el mantenimiento de una pequeña biblioteca (Altas, bajas, modificaciones, consultas, búsquedas por título y por autor). Puede ser útil para controlar el estado de su biblioteca particular.

Admite muchas mejoras, ya que está realizado a modo de ejemplo, por lo cual sólo se incluye en cada registro los siguientes campos: título, autor, editorial y observaciones, aunque se le pueden añadir otros, como materia, estante, precio, nº hojas, etc...

La forma de fichero es relativo, lo que permite la rápida localización de los datos y su modificación. Está realizado en el microordenador CBM 8032 de Commodore y la unidad de disco correspondiente.

Por no tratarse de un programa profesional, tanto éste como el fichero se encuentran en el mismo diskette, aunque con la oportuna modificación podrían separarse.



ELECCION DE LA OPERACION

Como hemos dicho anteriormente, el menú de este programa permite realizar las operaciones de Altas, Bajas, Modificaciones, Consultas. Búsqueda por título, Búsqueda por autor y una última operación para terminar el proceso. No obstante, se le podría dotar de otras operaciones diferentes, como puede ser aparición en pantalla de todos los libros, etc...

Al correr el programa aparecerá en pantalla el menú con todas las opciones posibles, debiendo elegir una de ellas. Los casos que se presentan son los siguientes:

ALTAS

Al elegir esta opción podremos introducir los datos referentes a nuevos libros que entren a formar parte de nuestra biblioteca, escribiendo el título del libro, su autor, la editorial y observaciones (aquí podemos introducir nuestra opi-

nión sobre el libro, la materia sobre la que trata, el estante en que se encuentra, etc.). Hay que tener en cuenta que el registro está creado con una longitud de 254 caracteres y por tanto, la suma de los caracteres de estos cuatro campos no debe sobrepasar los 254.

La forma de introducir los datos es abriendo el fichero y a continuación dar el número del registro que se desea usar para contener la información. El programa se encargará de ver si el registro existe o no (el fichero está dimensionado para 600 registros, este número se puede variar dependiendo de las características del ordenador) y también de si está ocupado ya o no dicho registro. En el caso de que esté ocupado no podremos darlo de alta allí, a no ser que antes demos de baja el contenido de ese registro.

Por último, tendremos la opción de repetir la operación, en cuyo caso podremos dar una nueva alta, y en caso contrario se cierra el fichero y vuelve al menú principal.

BAJAS

Esta opción nos permite borrar del fichero los datos de algún registro, por darse de baja o perder alguno de los libros.

Naturalmente, lo primero es abrir el fichero y decirle el número de registro a borrar; una

vez que se ha comprobado que dicho registro existe y que realmente está lleno, pregunta si estamos seguros de querer borrarlo; en el caso de que la respuesta sea *no*, aborta la operación, y si la respuesta es *sí*, entonces introduce un carácter blanco en ese registro.

A continuación y como vimos en el caso anterior, nos permite dos opciones: repetir la operación de dar otra baja o cerrar el fichero y volver al menú principal.

MODIFICACIONES

En el caso de elegir esta opción, podremos variar la información existente en uno, varios o todos los campos del registro deseado.

Tras abrir el fichero y dar un número de registro válido, aparece en pantalla la información contenida en ese registro. A continuación y situando el cursor en el lugar correspondiente, se podrá variar la información de los distintos campos.

Al igual que en los dos casos anteriores, se podrá elegir entre repetir la operación y cerrar el fichero y volver al menú.

CONSULTAS

Hace aparecer en pantalla la información contenida en el registro que se le indique.

```

2 REM PROGRAMADOR JOSE MANUEL RODRIGUEZ PROLONGO, 16-10-82
10 REM PROGRAMA DE PRUEBA. ARCHIVO RELATIVO DE UNA BIBLIOTECA
15 Z$=CHR$(13)
20 PRINT "Opciones"
30 PRINT "Opciones"
40 PRINT "Opciones"
50 PRINT "Opciones"
60 PRINT "Opciones"
70 PRINT "Opciones"
75 PRINT "Opciones"
77 PRINT "Opciones"
80 INPUT "Opciones"
90 IF OP$="A" THEN 200
100 IF OP$="B" THEN 400
110 IF OP$="C" THEN 600
120 IF OP$="D" THEN 800
125 IF OP$="E" THEN 2000
127 IF OP$="F" THEN 2100
130 IF OP$="G" THEN PRINT "Opciones"
135 PRINT "Opciones"
200 REM *****ALTAS*****
202 DOPEN #1,"BIBLIOTECA",L254,0
210 INPUT "Opciones"
220 IF R>600 OR R<1 THEN PRINT "Opciones"
222 RECORD#1,(R)
224 IF V$<>CHR$(255) THEN PRINT "Opciones"
230 RECORD#1,(R)
240 PRINT "Opciones"
250 PRINT "Opciones"
260 PRINT "Opciones"
270 PRINT "Opciones"
280 PRINT "Opciones"
281 GOSUB 1100
282 IF SI=1 THEN 210
290 DCLOSE#1
300 GOTO 20
400 REM *****BAJAS*****
402 DOPEN #1,"BIBLIOTECA",L254,0
410 INPUT "Opciones"
420 IF R>600 OR R<1 THEN PRINT "Opciones"
422 GOSUB 1000
424 IF VA=1 THEN 442
426 PRINT "Opciones"
427 IF SE$="N" THEN 442
428 IF SE$="S" THEN 426
430 RECORD#1,(R)
440 PRINT "Opciones"
442 GOSUB 1100
444 IF SI=1 THEN 410
450 DCLOSE#1
460 GOTO 20
600 REM *****MODIFICACIONES Y CONSULTAS*****
602 DOPEN #1,"BIBLIOTECA",L254,0
610 INPUT "Opciones"
620 IF R>600 OR R<1 THEN PRINT "Opciones"
622 GOSUB 1000
624 IF VA=1 THEN 748
630 RECORD#1,(R)
640 INPUT "Opciones"
650 PRINT "Opciones"
660 PRINT "Opciones"

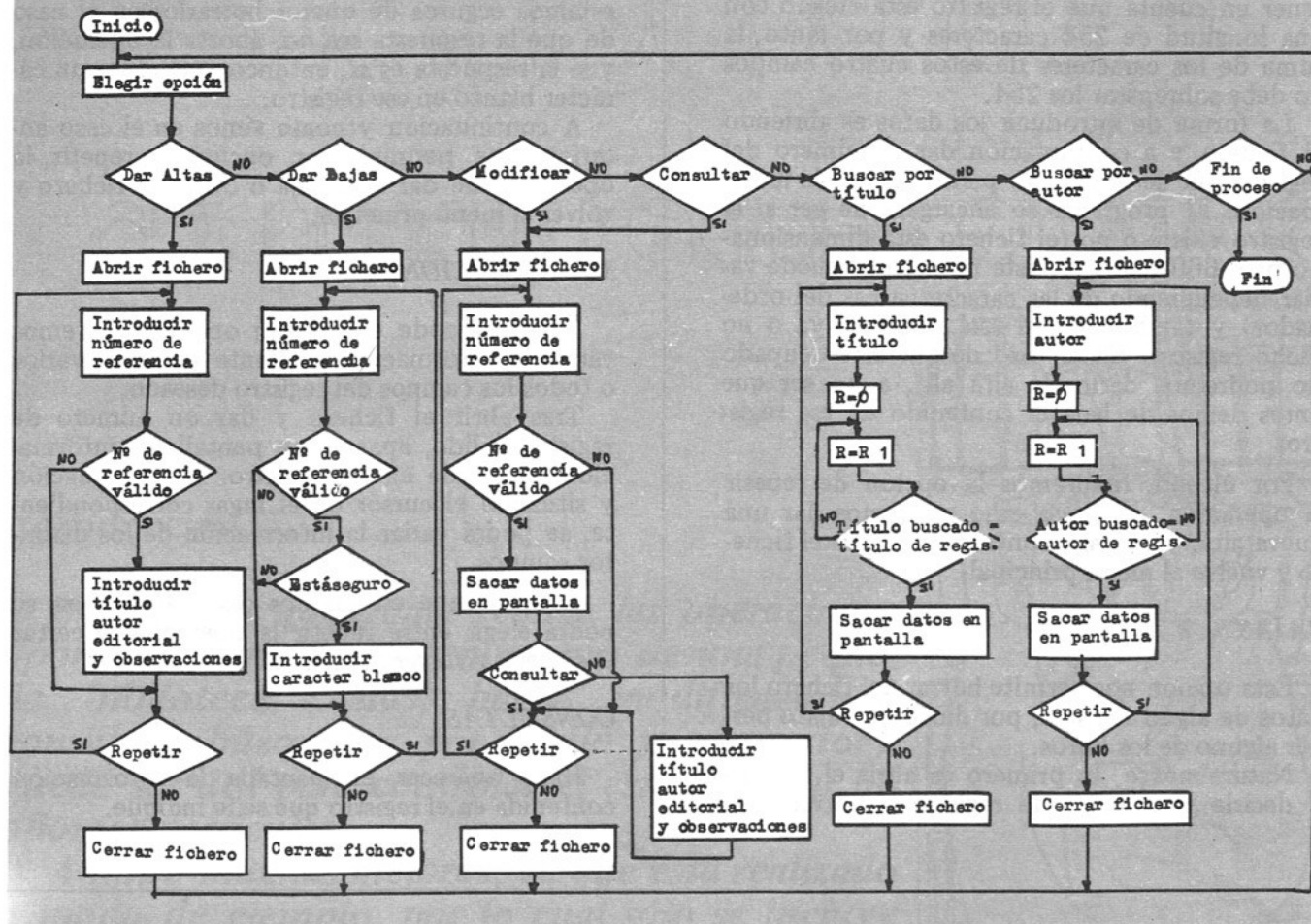
```

```

670 PRINT "Opciones"
680 PRINT "Opciones"
685 IF OP$="D" THEN 748
690 PRINT "Opciones"
700 PRINT "Opciones"
710 PRINT "Opciones"
720 PRINT "Opciones"
730 RECORD#1,(R)
740 PRINT "Opciones"
748 GOSUB 1100
749 IF SI=1 THEN 610
750 DCLOSE#1
760 GOTO 20
1000 REM Rutina de comprobar si esta
1010 VA=0:RECORD#1,(R)
1020 GET#1,V$
1030 IF V$<>CHR$(255) THEN PRINT "Opciones"
1040 RETURN
1050 VA=1:GOSUB 1200:RETURN
1100 REM Rutina de repetición
1110 PRINT "Opciones"
1120 SI=0:IF RE$="S" THEN SI=1:RETURN
1130 IF RE$<>"N" THEN 1100
1140 RETURN
1200 REM Rutina de tiempo
1210 FOR I=1 TO 3000:NEXT I:RETURN
2000 REM *****BUSQUEDA POR TITULO*****
2010 DOPEN #1,"BIBLIOTECA"
2020 R=0:X=0:INPUT "Opciones"
2030 R=R+1
2040 PRINT "Opciones"
2045 RECORD#1,(R)
2050 INPUT "Opciones"
2060 IF X$=A$ THEN X=X+1:GOSUB 2500
2070 IF R=600 THEN 2075
2074 GOTO 2030
2075 IF X=0 THEN PRINT "Opciones"
2076 GOSUB 1100
2077 IF SI=1 THEN 2020
2078 DCLOSE#1:GOTO 20
2100 REM *****BUSQUEDA POR AUTOR*****
2110 DOPEN #1,"BIBLIOTECA"
2120 R=0:X=0:INPUT "Opciones"
2130 R=R+1
2140 PRINT "Opciones"
2145 RECORD#1,(R)
2150 INPUT "Opciones"
2160 IF X$=A$ THEN X=X+1:GOSUB 2500
2170 IF R=600 THEN 2175
2174 GOTO 2130
2175 IF X=0 THEN PRINT "Opciones"
2176 GOSUB 1100
2177 IF SI=1 THEN 2120
2178 DCLOSE#1:GOTO 20
2500 PRINT "Opciones"
2510 PRINT "Opciones"
2520 PRINT "Opciones"
2530 PRINT "Opciones"
2540 PRINT "Opciones"
2550 GET E$:IF E$=" " THEN 2550
2560 RETURN

```

- ALGORITMO DEL PROGRAMA DE BIBLIOTECA -



Para ello, se le dice en primer lugar el número de registros a consultar, y tras comprobar su existencia y que tiene contenido, vuelca la información en pantalla.

Se puede volver a repetir la operación o cerrar el fichero y volver al menú, como en los casos anteriores.

BUSQUEDA POR TITULO

Esta opción permite buscar en el fichero determinado libro dando como referencia su título. Esto se hace comparando el título que se le indica con el de cada uno de los registros del fichero.

Para realizar esta operación, se abre el fichero y se introduce el título a buscar: a continuación un contador va dando los distintos valores a los registros, con objeto de comparar los títulos de cada uno de ellos con el indicado (compara el título de referencia con el del registro número uno; si no son iguales suma uno al contador y lo compara con el del registro número dos y así sucesivamente, hasta que coincida con alguno). Cuando se encuentra el título deseado, aparecen en pantalla todos los datos contenidos en ese registro, quedando el programa parado hasta pulsar cualquier tecla y entonces prosigue con la misma operación de búsqueda, a partir del registro en que estaba, por si hubiera otro libro con el mismo título. Así se llegaría hasta el último registro.

También aquí se puede volver a realizar la operación o cerrar fichero y volver al menú.

En cada momento aparece en pantalla un número, que irá variando y que indica por qué registro se va realizando la operación de búsqueda. Si no hubiera ningún libro con dicho título, también lo indicaría.

BUSQUEDA POR AUTOR

Al igual que en el caso anterior, esta opción permite obtener información del fichero, dando como referencia en este caso el nombre del autor.

Las operaciones realizadas son exactamente las mismas que en la búsqueda por título, sólo que comparando el autor indicado con el que contiene cada uno de los registros.

FIN DE PROCESO

Esta opción se utiliza cuando, una vez realizadas las distintas operaciones sobre el fichero, se desea salir del programa.

Borrar pantalla y dar fin al programa es la operación realizada por esta opción.

Todo lo aquí expuesto queda claramente reflejado en el Algoritmo del programa.

PROGRAMA

JOSE MANUEL RODRIGUEZ PROLONGO

Rally de Montecarlo para PC I500

Este es un programa que en dos versiones distintas, según la memoria disponible, y aprovechando las grandes posibilidades gráficas del PC-1500, simula un emocionante Rallye en el que entran en juego la habilidad y rapidez del piloto.

Antes de empezar a describir el juego, vamos a dar los requisitos: para la versión corta, basta con la memoria de base y para la versión extendida hace falta disponer de unos 2500 bytes (CE-151 ó CE-155).

Las diferencias entre las dos versiones son mínimas teniendo la versión extendida más efectos sonoros y la posibilidad de escoger dos niveles de dificultad.

DESCRIPCION DEL JUEGO.

La pantalla queda dividida en dos sectores: circuito y marcadores. Se trata pues de llevar el coche, que aparece en la izquierda de la pantalla, a la meta, centro de la pantalla. La zona del circuito está dividida a su vez en dos mitades escogidas al azar entre cinco posibles por nivel.

El marcador nos indica la velocidad, los puntos que tenemos y los coches de reserva que nos quedan.

El coche se representa por dos puntos consecutivos que recorren la pantalla. Se ha optado por no borrar la "estela" del coche por ganar rapidez y porque de esta forma se observa mejor la trayectoria del mismo.

MANEJO DEL COCHE.

Los controles son:

- "P" : para subir (girar a la izquierda)
- "L" : para bajar (girar a la derecha).
- "Q" : para acelerar.
- "A" : para reducir.
- " " : Freno de mano (espera un instante y deja el automóvil en la marcha más lenta).

DESCRIPCION DEL PROGRAMA

Línea	Comentario
1:	Archivo del record y el nombre del que lo posee.
10:	Inicialización del programa.
20:	Bucle de jugadores.
50:	Elección aleatoria y dibujo del recorrido.
60:	Se pinta la bandera de llegada.
70:	Se pintan los coches de reserva.
200:	Inicio del recorrido.
210:	Va a ver si ha habido choque o no.
220:	Entrada de instrucción.
230:	Acelerador.
240:	Freno.
250:	Freno de mano.
260:	Ascenso
270:	Con llamada a subrutina para ver si hay golpe o no.
290:	Fin del bucle del recorrido.
300:	Suma de puntos por final de recorrido.
310:	Comprueba si todavía nos queda algún coche.
320:	Si no pasa al siguiente jugador.
1000	Bucle que nos da quien ha ganado y si ha habido record.
1030	Se ejecuta si ha habido golpe: suena una sirena y nos quita un coche.
2000:	SUBROUTINA para ver si hay golpe o no, y si no lo hay nos suma puntos.
3000:	SUBROUTINA para archivar el record.
5000	
6000	
10.000	
	CIRCUITOS: 10000 - 10009 difícil NIVEL 0
	10010 - 10019 fácil NIVEL 1

Al principio de la partida el coche sale en la velocidad más rápida.

Hemos de hacer constar que los mandos no responden tan rápidamente como se desearía; y os dareis cuenta que se tardan unas vueltas partidas en cogerle el "truquillo".

Si se deja la tecla apretada, el coche sigue subiendo o bajando.

Si te estrellas una ambulancia vendrá a recogerte pero habrás perdido un coche.

PUNTUACION.

La puntuación viene dada por dos parámetros:

- La dificultad del trazado.
- El tiempo que se tarda en hacer el recorrido.

La fórmula es así:

por cada avance del coche:

$$P = P + (\text{dificultad del semitrazado } 1 - \text{dificultad del semitrazado } 2) * \text{NIVEL.}$$

y por cada trazado recorrido

$$P = OP + INT(1/TIME * 0'04).$$

Una vez finalizada la competición, la máquina nos da el vencedor y su puntuación; y en caso de que la obtenga record, nos pedirá el nombre y lo archivará junto con la puntuación; pero de una forma un tanto original que explicaremos más adelante.

Una vez dada una idea general de como funciona el programa (ver pág. 1), vamos a describir más detalladamente las subrutinas 3000 y 5000 porque creemos que tienen especial interés.

SUBROUTINA 3000.

Es la subrutina que controla en cada avance si nos hemos salido de la carretera.

Las variables A4 y A5, tienen el valor del POINT de la primera y de la segunda línea que caen en la vertical del coche pero sin tener en cuenta a este último. H y H1, tienen el mismo valor pero ya tienen en cuenta la posición del coche.

Luego restando respectivamente H y A4 y H1 y A5 y si el resultado de alguna de ellas es igual a 0 esto implica que nos hemos salido de la carretera, bifurcando a la subrutina 2000.

En el caso de que ninguna sea 0, se procede a sumar los puntos correspondientes.

SUBROUTINA 5000.

Con los descubrimientos que se están haciendo sobre el interior de la PC-1500, ya podemos conocer como se guarda un programa y donde (ver en nº 15 CONFIDENCIAS DEL PC-1500 (II)). Y aprovechándose de esto hemos diseñado una ingeniosa forma de guardar datos de modo que no se nos borren al hacer un CLEAR; es decir modificando mediante programa una línea de este (en nuestro caso la primera línea).

El método siguiente:

La primera empieza en la dirección 40C5; los dos primeros bytes son para el número de línea; el siguiente es para longitud de ésta y después de línea en

sí. Pues bien si en los huecos que hemos reservado.,

1: B\$ = " - - - 16 caracteres" : C\$ = 5 caracteres".

que van B\$ desde & 40CE hasta & 40DE y C\$ desde &40E4 hasta &40E9, vamos introduciendo (mediante POKE) los códigos ASCII de los caracteres deseados, hallaremos con cierta sorpresa que la línea de programa se ha modificado actualizando mediante el siguiente bucle el record y el nombre del poseedor de este.

5000 Pide el nombre.

5010 Da valores a los parámetros del bucle y va a la subrutina 6000.

6000 Toma uno a uno los dos caracteres del nombre y del record; los traduce a código ASCII y los poka en la dirección deseada.

INICIALIZACION DEL PROGRAMA

- Paso nº 1 RUN
- Paso nº 2 Aparece el record y el nombre del que lo posee.
- Paso nº 3 ENTER.
- Paso nº 4 Aparece "NUMERO DE JUGADORES".
- Paso nº 5 Introducir este número (cuidado con la memoria que queda).
- Paso nº 6 ENTER
- Paso nº 7 Aparece "JUEGA EL 1".
- Paso nº 8 ENTER
- Paso nº 9 Empieza la carrera.

LISTA DE VARIABLES

- ALFANUMERICAS.

- A\$: Instrucción de dirección.
- B\$: Nombre del recordman.
- c\$: Record.

- NUMERICAS.

- A NIVEL.
- A4,A5 POINT sin el coche.
- B Posición en el eje Y.
- C Coches que nos quedan.
- D Direcciones para la subrutina 6000.
- G Puntuación del ganador.
- H,H1 POINT con el coche.
- I Posición en el eje X, e índice de varios bucles.
- J Jugador.
- L Auxiliar: para pintar los coches que quedan.
- M Variable para pintar la pista.
- R Para las READ de la música.
- S
- V Velocidad.
- X Índice del bucle de sirena, longitud de los datos en la subrutina 6000.

- MATRICIALES.

- P (J) Puntos de cada jugador.
- S (M) Trozos de la pista.


```

1: CLEAR : B$=""
: C$=""
2: PRINT B$;" CON
";C$
* 3: DATA 87, 125, 66
, 162, 87, 125, 66
, 162, 93, 110
* 4: DATA 76, 140, 93
, 110, 76, 140, 10
5, 105, 87, 125
* 5: DATA 87, 125, 76
, 140, 87, 125, 10
5, 105, 117, 97
* 6: DATA 87, 125, 93
, 110, 105, 105, 9
3, 110, 105, 105
* 7: DATA 117, 97, 10
5, 105, 93, 110, 1
05, 105, 117, 97
10: INPUT "NUMERO
DE JUGADORES";
N: DIM P(N-1), S
(1)
20: FOR J=0 TO N-1
* 25: FOR I=1 TO 15:
READ R, S: BEEP
1, R, S: NEXT I
30: CLS : PRINT "JU
EGA EL "; J+1:
CLS
40: WAIT 0: L=85, C=
5, A=0
* 45: INPUT "NIVEL (1
-FACIL; 0-DIFIC
IL)"; A
50: RANDOM : FOR M=
0 TO 1: S(M)=RND
5: GOSUB 10000+
A*10+2*(S(M)-1
): NEXT M
60: A4=63, A5=63:
GCURSOR 80:
GPRINT "55AA55
AA55";
70: B=8: GCURSOR 0:
H=71, H1=H:
GCURSOR 90:
GPRINT L; L;:
CURSOR 18:
PRINT U; P(J)
200: BEEP 5, 30, 60:
TIME =0: FOR I=
0 TO 75 STEP 2
210: GOSUB 3000
220: A$=INKEY$
230: IF A$="Q" LET U
=U+1: IF U<0 LET
U=0: WAIT U
240: IF A$="A" LET U
=U+1: IF U>5 LET
U=5: WAIT U
250: IF A$=" " WAIT
20: PRINT : U=5:
WAIT 5
260: IF A$="P" LET B
=INT ((B+1)/2)
: IF B<1 LET B=1
: GOSUB 3000
270: IF A$="L" LET B
=B*2: IF B>64
LET B=64: GOSUB
3000
280: CURSOR 18:
PRINT U; P(J)
290: NEXT I
300: P(J)=P(J)+INT
(1/TIME *.04):
WAIT 200:
CURSOR 18:
PRINT U; P(J):
WAIT 0
310: IF C>0 GOTO 50
* 315: RESTORE 5:
RESTORE 4:
RESTORE 3
320: NEXT J
1000: FOR J=0 TO N-
1: IF J=0 LET
G=P(0), N=0
1010: IF P(J)>G
LET G=P(J): N
=J
1020: NEXT J: CLS :
WAIT 150:
PRINT "GANA
EL "; N+1; " C
ON"; G; " PUNT
OS"
1030: IF G>VAL C$
BEEP 3: PRINT
"CON NUEVO R
ECORD": GOTO
5000
1140: END
2000: FOR X=0 TO 3:
BEEP 1, 80, 25
0: BEEP 1, 140
, 250: NEXT X
2010: GCURSOR (I+2
): GPRINT 0; 0
: C=C-1:
GCURSOR 90:
IF C=4 LET L=
21
2020: IF C=3 LET L=
5
2030: IF C=2 LET L=
1
2040: IF C=1 LET L=
0
2050: IF C=0 WAIT :
GOTO 310
2060: GPRINT L; L;:
RETURN
3000: H=POINT IOR
B, H1=POINT (
I+1) OR B:
GCURSOR I
3010: GPRINT H; H1;
3030: IF ((H-A4)=
0) OR ((H1-A5
)=0) GOSUB 2
000: RETURN
3040: P(J)=P(J)+(S
(0)+S(1))*C
NOT A+3), A4=
POINT (I+2),
A5=POINT (I+
3): RETURN
* 5000: RESTORE 7:
RESTORE 6
* 5010: FOR I=1 TO 10
: READ R, S:
BEEP 1, R, S:
NEXT I
* 5020: INPUT "DIME
TU NOMBRE"; B
$
5030: D=&40CE: X=16
: GOSUB 6000:
D=D+22: X=5: B
$=STR$ G:
GOSUB 6000:
BEEP 1: END
6000: FOR I=1 TO X:
POKE D+I, ASC
MID$ (B$, I, 1
): NEXT I:
RETURN
10000: GPRINT "6767
63636B636363
63637B7B6363
6363636F6F63
636363637B7B
63636363636F
6F63";
10001: GPRINT "6363
637373";:
RETURN
10002: GPRINT "7373
717171797D7D
797171736363
63636747474F
4F4F4F5F4F4F
474363737B63
6363";
10003: GPRINT "6767
677373";:
RETURN
10004: GPRINT "7363
6363676F6363
63637B717171
7D7979716161
4343474F4343
434B4B4B7363
6767";
10005: GPRINT "6767
676767";:
RETURN
10006: GPRINT "6767
677371717D7D
716163634747
474F4F5F5F4F
4F4747677747
47475F474363
6373";
10007: GPRINT "7B73
737373";:
RETURN
10008: GPRINT "6363
637B4343434F
41414141797D
71614143434F
4F5F47434363
73797979797D
7961";
10009: GPRINT "6167
676777";:
RETURN
10010: GPRINT "4141
414141414149
494141414141
414141416363
636347474747
636363634141
41";
10011: GPRINT "4141
41414141";:
RETURN
10012: GPRINT "4141
414141416161
616171797971
616161614141
414143434347
474743434141
41";
10013: GPRINT "4141
41414141";:
RETURN
10014: GPRINT "4141
414141414161
616141414147
47474F4F4F4F
4F4747474743
436161717171
71";
10015: GPRINT "7979
79716141";:
RETURN
10016: GPRINT "4141
414141494949
414141616171
717171616167
676363636363
636363776363
63";
10017: GPRINT "6363
63636363";:
RETURN
10018: GPRINT "6767
63636B636363
63637B7B6363
6363636F6F63
636363637B7B
63636363636F
6F63";
10019: GPRINT "6363
637373";:
RETURN

```

NOTA: MODIFICACIONES PARA LA VERSION 1K8.

- SUPRIMIR LAS SENTENCIAS MARCADAS CON *.
- ESCOGER un paquete de sentencias 10000 - 10009 ó 10010 - 10019: pero escoja la que escoja póngale los números de sentencia de la 10000 - 10009.
- Poner la sentencia 5000: INPUT "DIME TU NOMBRE"; B\$.

NOTA A LOS USUARIOS DE ESTE PROGRAMA:

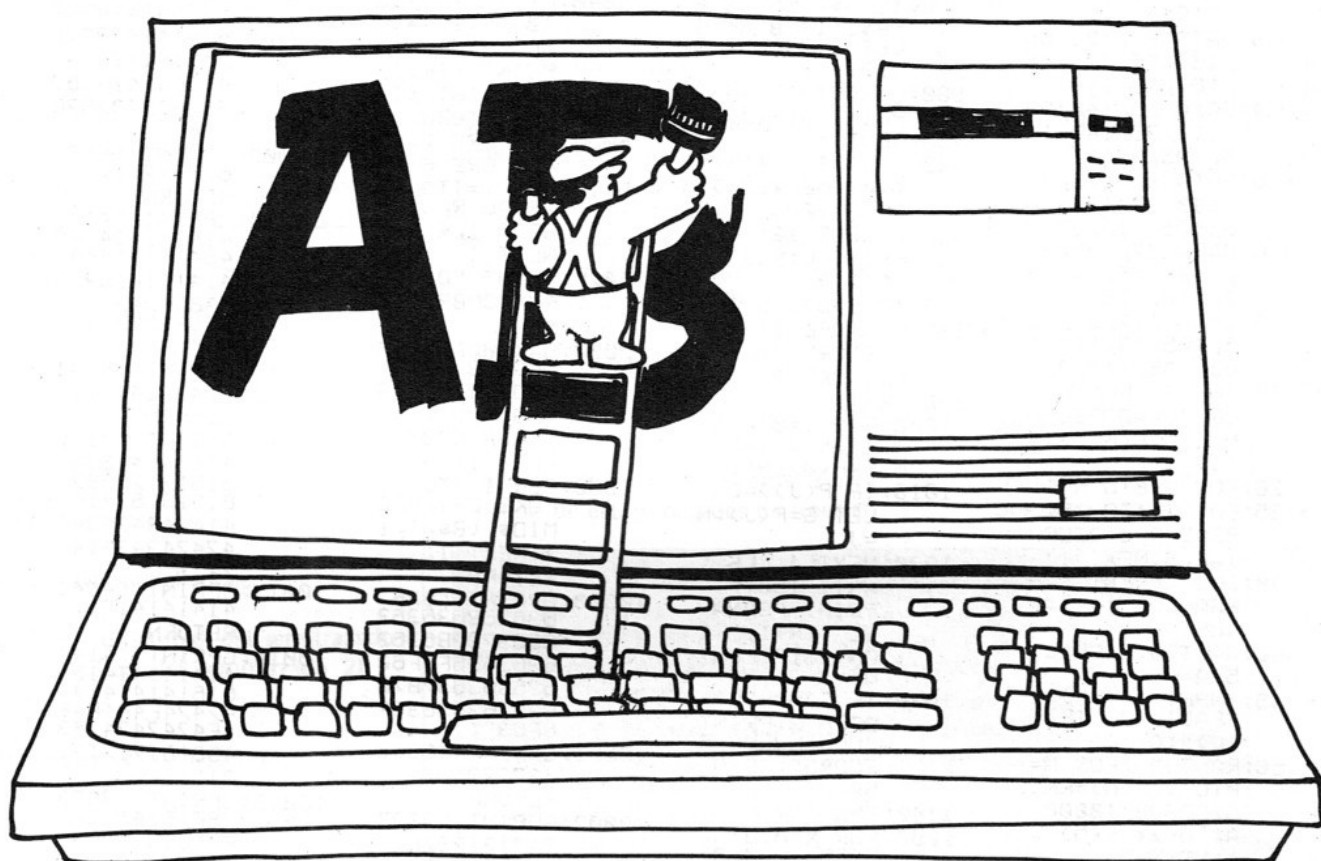
Para los principiantes es recomendable empezar por el nivel 1 (Fácil) y

acostumbrarse al manejo de los mandos.

Para los expertos, es decir, después de haberse echado unas cuantas partidas y si quiere que el coche responda antes y marche más rápido sugiero que quite los controles "A", "Q"; " " ; ya

que cuando se consigue un cierto nivel no sirven para nada.

*Iñaki Cabrera.
Victor Manuel Díaz.*



BIG-PATTERN (Gran silueta)

La finalidad del programa BIG-PATTERN es, en principio, la de poder visualizar en pantalla mensajes en grandes caracteres (o en impresora) para su uso, como publicidad en escaparates, demostraciones, etc. . . ; ahora bien, las aplicaciones que se puedan derivar resultan numerosas.

El programa incluye los siguientes procesos:

1.- DEFINICION DE LOS CARACTERES: Cada carácter "gigante" está constituido por una matriz de 6 x 7 caracteres convencionales. Esta opción permite definir cuáles de los puntos de esta matriz se deben de "iluminar" para conseguir cada carácter.

Como "punto" se puede escoger cualquiera de los caracteres alfanuméricos o gráficos que disponga el ordenador; para cada letra o carácter gigante se puede definir un tipo diferente de punto.

La aplicación más usual es definir cada carácter gigante con cada uno de los caracteres normales disponibles, en gran escala; aunque se puede definir cualquier gráfico.

2.— GRABACION EN ARCHIVO: Podemos grabar los caracteres definidos en un diskette ó cassette, evitando el definirlos cada vez que carguemos el programa. En el microordenador con que se ha realizado el programa (SHARP MZ 80-B) los nombres de archivo permitidos son de hasta 16 caracteres.

3.— LECTURA ARCHIVO: Lee el archivo de caracteres especificado y lo carga en memoria máquina.

4.— CONSULTA EN PANTALLA: Indicando los códigos ASCII de inicio y fin de la consulta, visualiza en pantalla el carácter "gigante" asociado a cada uno de ellos; esta opción es útil para comprobar los caracteres definidos.

5.— MENSAJES EN IMPRESORA: Una vez entrados los mensajes (de 1 a 255) se codifican a "carácter gigante" y se imprimen por impresora la cantidad de originales solicitados. Se puede seleccionar con espacio entre líneas

o sin él, y el número de líneas de separación entre cada línea de mensajes.

6.— MENSAJES EN PANTALLA: Opción similar a la 5, pero por pantalla.

7.— CAMBIO DE PUNTO: Una vez definidos los caracteres, se puede cambiar el carácter del "punto" de que están formados. Seleccionando los márgenes de inicio/fin del proceso aparecen, a modo de control, los caracteres que se van tratando.

El listado corresponde a la versión del programa para el microordenador SHARP MZ 80 B (Basic SB 6510). Las sentencias utilizadas son fácilmente "traducibles" a otros tipos de BASIC (Microsoft, Comodore, etc.). Puede resultar interesante, asimismo, convertir el programa para mod gráfico de media o alta resolución.

Otra interesante posibilidad para los lectores inquietos sería crear un archivo secuencial para almacenar grupos de mensajes.

El número máximo de caracteres por mensaje (con pantalla de 80 carac. y matriz de carácter gigante de 6 x 7) es de 9 por línea.

El programa necesita, aproximadamente, 12 Ko. de memoria (ejecutado y con 45 caracteres definidos).

Para mayor comprensión del programa y facilitar su conversión, os detallo algunas de las instrucciones más particulares con una breve explicación de las mismas:

— **PRINT CHR\$(6)** = El código ASCII 6 corresponde al borrado de pantalla y cursor a posición HOME. (equivalentes = **CLS/ERASE/PRINT** "☐").

— **PRINT CHR\$(5)** = Corresponde a la instrucción **CURSOR HOME**, sin borrado de pantalla. (Equivalente: **PRINT** "☐"/**PRINT CHR\$(n)**, donde n es el código correspondiente y varía según el ordenador).

— **CURSOR x,y** = Siendo 'x' e 'y' número enteros, variables numéricas, expresiones ó combinación de ellos. Coloca el cursor en la posición indicada por "x"

— Eje horizontal e "y"— Eje vertical. (Equivalente a: **LOCATE x,y/PRINT@n/PRINT TAB (x,y)/...**)

— **STRING\$("X",n)** = Visualiza o imprime una cadena de n elementos "X" ("X" puede substituirse por una expresión del tipo **CHR\$(x)**) Equivale: **STRING\$(n,"X"),...**

— **CONSOLE** (expresión) = Es una instrucción de control de formato de pantalla, las variables son:

— **CONSOLE C40:** Coloca pantalla en modo de 40 caracteres x 24 líneas.

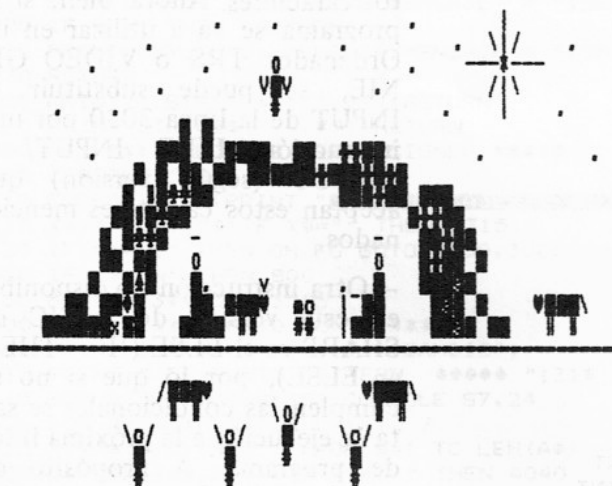
— **CONSOLE C80:** Coloca pantalla en modo de 80 caracteres x 24 líneas. (Estas dos equivalen a = **WIDTH 80 / PRINT CHR\$(n)**, siendo n el código correspondiente).

— **CONSOLE S a,b:** Define la zona de "scroll" programable entre las líneas a y b. (Nota: Esta instrucción no es imprescindible el programa propuesto. Si solamente, para conseguir una mejor presentación o ahorrar instrucciones más complejas).

— **GET X\$** = Acepta un carácter alfanumérico, asociando el valor de nulo si no se ha pulsado ninguna tecla. (Equivale = **X\$ = INKEY\$ / X\$ = KIN\$. . .**)

Ejecución de Prueba.

PAUL AMOR



```

1 REM *****
2 REM *** B I G - P A T T E R N ***
3 REM *****
4 REM *** AUTOR = J.M. Vidal Lacasa ***
5 REM *****
6 REM *** COPYRIGHT EL AUTOR Y ***
7 REM *** EL ORDENADOR PERSONAL ***
8 REM *****
9 REM
10 DIM A$(255,7),S$(6),M(20),Z$(39)
15 Z$="+B1R0+B1R0+B1R0":TEMPO 7
20 Z1$=STRING$(6,40):GOTO 800
100 REM ***** DEF. CARACTERES *****
105 PRINT CHR$(6);" ***** DEF. ";
110 PRINT "CARACTERES BIGPATTERN ***** ";Z1$
115 GOSUB 300:GOSUB 350:N=IN:CONSOLE S 7,24
120 PRINT CHR$(6);" 1 = * * O = ESP. *";
125 PRINT " CR = CORREC.":PRINT
130 PRINT " ASCII = ";N;" VAR. = ";
135 PRINT CHR$(N);" ":CURSOR 0,12
140 FOR Q=10 TO 16:PRINT TAB(15);".....":NEXT Q
145 FOR L=1 TO 7:CURSOR 15,(11+L):GOSUB 450:NEXT L
150 N=N+1:IF N>F THEN 160
155 GOTO 120
160 CONSOLE S 0,24:GOTO 800
250 REM ***** ENTRADA NOMBRE ARCHIVO *****
255 CURSOR 0,10:INPUT "NOMBRE ARCHIVO (16 c.) = ";NA$
260 IF LEN(NA$)<16 THEN RETURN
265 MUSIC Z3$:CURSOR 0,10
270 PRINT STRING$(6,80):GOTO 250
300 REM ***** ENTRADA INICIO/FIN *****
305 CURSOR 0,4:INPUT "Cod. ASCII INICIO = ";IN
310 INPUT "Cod. ASCII FIN = ";F
315 IF (F>255) + (IN>255) THEN 330
320 IF (F<IN) + (IN<31) THEN 330
325 RETURN
330 MUSIC Z3$:GOTO 300
350 REM ***** ENTRADA ASCII PUNTO *****
355 CURSOR 0,10:INPUT "CODIGO ASCII PUNTO = ";CA
360 IF (CA<31)+(CA>255) THEN 350
365 PRINT:PRINT CA;" CORRESPONDE A ";CHR$(CA):PRINT
370 PRINT " CORRECTO (S/N) ? "
375 X$="":GET X$:IF X$="" THEN 375
380 IF X$="S" THEN RETURN
385 IF X$="N" THEN CURSOR 0,10:PRINT STRING$(6,39):GOTO 350
390 GOTO 375
400 X$="":GET X$:IF X$="" THEN 400
405 X=ASC(X$):IF X=49 THEN C$=CHR$(CA):GOTO 425
410 IF X=48 THEN C$=" ":GOTO 425
415 IF X=13 THEN S=S-2:PRINT CHR$(4);:RETURN
420 GOTO 400
425 PRINT C$;:RETURN
450 REM ***** ENTRADA LINEA DEF. *****
455 S$="":FOR S=1 TO 6:GOSUB 400:S$(S)=C$:NEXT S
460 S$=S$(1)+S$(2)+S$(3)+S$(4)+S$(5)+S$(6)
465 A$(N,L)=S$:PRINT:RETURN
500 REM ***** GRABACION ARCHIVO *****
505 PRINT CHR$(6);" ***** GRABACION ARCHIVO";
510 PRINT " BIGPATTERN ***** ";Z1$
515 GOSUB 250:WOPEN #1,FD1,NA$
520 FOR A=31 TO 160:FOR N=1 TO 7
525 PRINT #1,A$(A,N):NEXT N:NEXT A
530 CLOSE:GOTO 800
600 REM ***** LECTURA ARCHIVO *****
605 PRINT CHR$(6);" ***** LECTURA ";
610 PRINT "ARCHIVO BIGPATTERN ***** ";Z1$
615 GOSUB 250:ROPEN #1,FD1,NA$
620 FOR A=31 TO 160:FOR N=1 TO 7
625 INPUT #1,A$(A,N):NEXT N:NEXT A
630 CLOSE:GOTO 800
800 REM ***** MENU GENERAL *****
805 CONSOLE S0,24,C40:FG=0
810 PRINT " B I G - ";
815 PRINT "P A T T E R N * * * * *":PRINT Z1$
820 CURSOR 5,5:PRINT "1 - DEFINICION GRAFOS":PRINT
825 PRINT TAB(5);"2 - GRABACION ARCHIVO DEF.":PRINT
830 PRINT TAB(5);"3 - LECTURA ARCHIVO DEF.":PRINT
835 PRINT TAB(5);"4 - VISUALIZACION PANTALLA":PRINT
840 PRINT TAB(5);"5 - IMPRESION PAPEL":PRINT
845 PRINT TAB(5);"6 - MENSAJES PANTALLA":PRINT

```

– Instrucciones “musicales”: TEMPO 7 = define el “tempo” o velocidad de ejecución de la música. / MUSIC Z3\$ = Suena un pitido de nota, escala y duración definidas por la variable Z3\$ (Ver línea 15 del listado).

La instrucción MUSIC Z3\$ no es imprescindible, provoca la ejecución de 3 pitidos cortos a modo de advertencia si se ha efectuado una entrada incorrecta. En la máquinas que lo dispongan se puede suplir por BEEP 3.

– GET X = Similar a GET X\$, pero con la diferencia de que solamente acepta caracteres numéricos.

– PRINT/P = Instrucción de impresión en impresora. (Equivalente a LPRINT). Los códigos de comando de impresora del SHARP MZ 80 B utilizados en este programa, son:

– PRINT/P CHR\$(16): Comando, de impresión en modo de separación entre línea y línea.

– PRINT/P CHR\$(17): Impresión sin separación entre línea y línea.

– PRINT/P CHR\$(5): PAPER HOME – Coloca el papel al principio de página.

– Otro código utilizado es el CHR\$(4) (PRINT CHR\$(4)); provoca el desplazamiento del cursor un carácter a la izquierda.

NOTA: La instrucción INPUT de la versión citada de SHARP no admite espacios a la izquierda, comas, ni comillas. Ello disminuye las posibilidades de entrada de cartel al no poder disponer de estos caracteres. Ahora bien, si el programa se va a utilizar en un Ordenador TRS ó VIDEO GENIE, se puede substituir el INPUT de la línea 3030 por una instrucción LINE INPUT/ o LINPUT (según versión) que aceptan estos caracteres mencionados.

– Otra instrucción no disponible en esta versión de BASIC de SHARP es el ELSE (IF – THEN – ELSE), por lo que si no se cumplen las condicionales se salta la ejecución a la próxima línea de programa. A propósito de condicionales, los operadores lógicos utilizados son: * (AND) y + (OR) (ejemplo: la línea (IF (A=0) + (A=1) THEN 100


```

850 PRINT TAB(5);" - CAMBIO PUNTO"
855 CURSOR 7,21:PRINT " SELECCIONE OPCION "
860 X=0:GET X:IF X=0 THEN 860
865 ON X GOTO 100,500,600,1000,2000,3000,4000
870 MUSIC Z3$:GOTO 860
1000 REM ***** CONSULTA PANTALLA *****
1005 PRINT CHR$(6);" **** CONSULTA ";
1010 PRINT "CARACTERES PANTALLA ****";Z1$
1015 CONSOLE S4,24:GOSUB 300
1020 FOR A=IN TO F:PRINT CHR$(6);" ** ASCII = ";A;
1025 PRINT " ** CHARACTER = ";CHR$(A);" **"
1030 PRINT:PRINT:PRINT:PRINT
1035 FOR L=1 TO 7:PRINT TAB(15);A$(A,L)
1040 NEXT L:PRINT:PRINT
1045 PRINT TAB(5);" Pulse cualquier tecla..."
1050 PRINT TAB(8);" (F1 = Finalizar)"
1055 X$="":GET X$:IF X$="" THEN 1055
1060 IF ASC(X$)=16 THEN A=F
1065 NEXT A:CONSOLE S0,24:GOTO 800
2000 REM ***** MENSAJES IMPRESORA *****
2005 PRINT CHR$(6);" **** IMPRESION";
2010 PRINT " MENSAJES PAPEL ****"
2015 PRINT Z1$:CONSOLE S3,24
2020 CURSOR 0,5
2025 INPUT "SEPARACION ENTRE LINEAS (S/N) ? ";S$
2030 IF S$="S" THEN PRINT/PCHR$(16):GOTO 2045
2035 IF S$="N" THEN PRINT/PCHR$(17):GOTO 2045
2040 MUSIC Z3$:GOTO 2020
2045 FG=1:PRINT:GOTO 3020
3000 REM ***** MENSAJES PANTALLA *****
3005 PRINT CHR$(6);" *** VISUALIZACION";
3010 PRINT " MENSAJES PANTALLA ***";Z1$
3015 FG=2
3020 PRINT " MAX.40 ENTRADAS * FINALIZAR = FIN"
3025 PRINT:FOR J=0 TO 20
3030 INPUT "CARTEL = ";Z$:Z$=LEFT$(Z$,9)
3035 IF Z$="FIN" THEN NM=J-1:J=20:GOTO 3045
3040 Z$(J)=Z$:NM=J
3045 NEXT J:GOSUB 3200
3055 CONSOLE C80:PRINT CHR$(6);
3060 FOR V=1 TO NR:FOR ME=0 TO NM:Z$=Z$(ME):LG=LEN(Z$)
3065 TA=(79-(LG*6)+(LG-1)*3))/2
3070 FOR X=1 TO LG:M(X)=ASC(MID$(Z$,X,1)):NEXT X
3075 FOR N=1 TO 7:T=TA:FOR X=1 TO LG
3080 IF FG=1 THEN PRINT/P TAB(T);A$(M(X),N);:GOTO 3090
3085 PRINT TAB(T);A$(M(X),N);
3090 T=T+9:NEXT X:IF FG=1 THEN PRINT/P:GOTO 3100
3095 PRINT
3100 NEXT N:FOR L=1 TO SE
3105 IF FG=1 THEN PRINT/P STRING$(" ",78):GOTO 3115
3110 PRINT STRING$(" ",78):FOR S=1 TO 150:NEXT S
3115 NEXT L:NEXT ME:FOR Q=1 TO SM
3120 ON FG GOSUB 3250,3260:NEXT Q
3125 NEXT V:GOTO 3300
3200 REM ***** CONDICIONES *****
3205 PRINT:INPUT "No. REPETICIONES = ";NR
3210 PRINT:PRINT "ENTRE LA SEPARACION ENTRE : "
3215 PRINT:INPUT "...LINEAS MENSAJE = ";SE
3220 IF NR=1 THEN RETURN
3225 PRINT:INPUT "...MENSAJE/MENSAJE = ";SM
3230 RETURN
3250 PRINT/P STRING$(" ",79):RETURN
3260 PRINT STRING$(" ",79):RETURN
3300 REM ***** NO MAS REPETICIONES *****
3305 CONSOLE C40
3310 CURSOR 8,10:PRINT " MAS MENSAJES ? (S/N) "
3315 X$="":GET X$:IF X$="" THEN 3315
3320 IF X$="S" THEN ON FG GOTO 2000,3000
3325 IF X$="N" THEN 800
3330 MUSIC Z3$:GOTO 3315
4000 REM ***** CAMBIO PUNTO *****
4005 PRINT CHR$(6);" ***** CAMBIO";
4010 PRINT " DE PUNTO PATTERN ***** ";Z1$
4015 GOSUB 300:GOSUB 350:CONSOLE S7,24
4020 FOR A=IN TO F:FOR L=1 TO 7
4025 A$=A$(A,L):S$=" ":FOR W=1 TO LEN(A$)
4030 C$=MID$(A$,W,1):IF C$=" " THEN 4040
4035 S$=S$+CHR$(CA):GOTO 4045
4040 S$=S$+" "
4045 NEXT W:A$(A,L)=S$:NEXT L
4050 PRINT CHR$(A);" - ";
4055 NEXT A:CONSOLE S0,24:GOTO 800

```

equivale a: (IF (A=0) OR (A=1) THEN 100.

INSTRUCCIONES DE MANEJO DE ARCHIVOS (Acceso secuencial, que es el utilizado en el programa):

– WOPEN #1, FD1, NAS = (Write Open) Abre el archivo de acceso secuencial al modo de grabación; asignando el canal, o Número lógico 1 (#1), la unidad de diskettes 1 (FD1), y el nombre de archivo el correspondiente a la variable NAS.

– ROPEN # 1, FD1, NAS = (Read Open) Abre el archivo de acceso secuencial, en modo de lectura de datos. (Iguales características de número lógico y unidad de discos).

– PRINT #1, A\$ = Graba en el archivo secuencial (que debe de haber sido abierto con anterioridad) asignado al canal ó número log. 1 (#1) la variable de string A\$.

– INPUT #1,A\$ = Lee del archivo secuencial asignado a #1 la variable A\$.

– CLOSE #1 = Cierra el archivo asignado a #1.

– CLOSE = Sin indicación de número lógico, cierra todos los archivos abiertos.

Como última observación, y para la conversión a equipos cuyo formato de pantalla no disponga de 80 caracteres por línea, la línea 3030 contiene la instrucción Z\$=LEFT\$(Z\$,9), se debe cambiar el 9 por el número de caracteres que admita la pantalla (refiriéndome a caracteres "gigantes" de 6 x 7 caracteres normales). Teniendo en cuenta la separación entre carácter y carácter, que es de 3 caracteres convencionales.

No hace falta mencionar las posibilidades del programa con un ordenador que disponga de gráficos de alta resolución y varios colores. Espero recibir noticias de lectores que realicen algunas de estas modificaciones, así como las consultas de los problemas que os pudieran surgir de la introducción del programa. □

José María Vidal Lacasa.
Barcelona, Octubre 1982.

A hacer chuletas

Bernard Damman

Los shadock son unos pájaros extraterrestres cuyo lenguaje es de lo más abstracto. He aquí un pequeño traductor cuya eficacia no admite demostración...

El lenguaje shadock sólo dispone de cuatro palabras: «GA, BU, ZO y MEU», a las que añadiremos el silencio o espacio (« »). Esta economía de palabras no dice nada sobre el carácter superevolucionado del pensamiento de estos animales; podemos conver-

cernos de ello leyendo un verso de uno de sus poetas, o una fórmula matemática de vanguardia de uno de sus sabios.

El programa realiza la traducción en los dos sentidos. La traducción del español al shadock no entraña ninguna dificultad. La

traducción en el otro sentido requerirá la atención para introducir el texto shadock sin error.

En particular, los espacios deberán ser respetados tanto en número como en emplazamiento, incluidos los del principio y final de texto (los asteriscos que encuadran el texto shadock sirven para eso, no formando parte del texto a introducir). Todos los espacios son «significativos», y el lenguaje shadock los tiene en cuenta.

La impresora será útil para facilitar la traducción, pero no es indispensable. Al finalizar el programa, puede obtenerse la impresión del texto apretando la tecla C (o haciendo NEWLINE). El programa permitirá la codificación de mensajes secretos.

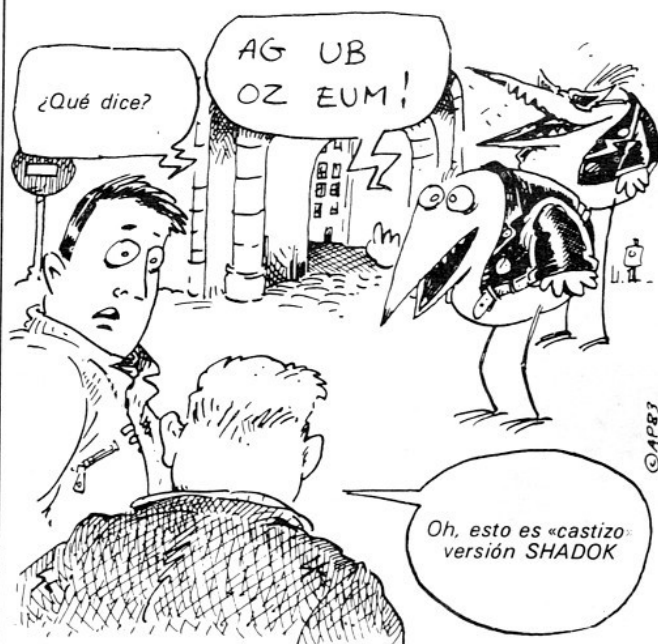
Ciento veinte dialectos para guardar vuestros secretos

Para asegurar el secreto, es suficiente emplear uno de los ciento veinte dialectos shadock. El «GABUZOMEU», utilizado aquí, es el más empleado. Para utilizar, por ejemplo; el «BUZO-GAMEU» no hay más que cambiar el orden de los cinco números N=0, 1, 2, 3, 4 por N=2, 3, 0, 1, 4 en las líneas 300 a 340 y 720 a 790. En este último, MEUZOGA-BU será ZAGOMUBEU, ¿no es cierto?



© AP83

PROGRAMA SHADOCK



```

2 CLS
5 GOTO 20
10 SAVE "SHADOK"
20 PRINT TAB 8;"TRADUCTOR"
30 PRINT AT 10,0;"1 CASTELLANO
SHADOK"
40 PRINT "2 SHADOK-CASTELLANO"
50 PRINT AT 21,0;"INTRODUCIR E
L NUMERO"
60 INPUT R$
65 CLS
70 IF R$="1" THEN GOTO 1000
80 IF R$="2" THEN GOTO 1500
90 IF R$="0" THEN STOP
95 GOTO 20
1000 REM TRADUCE A$(CASTELLANO)
A$SHADOK(B$)
110 REM
120 LET B$=""
130 FOR I=1 TO LEN A$
140 LET C=CODE (A$(I))
150 LET N1=INT ((C/25))
160 LET N2=INT ((C-25*N1)/5)
170 LET N3=C-25*N1-5*N2
180 LET N=N1
190 GOSUB 300
200 LET N=N2
210 GOSUB 300
220 LET N=N3
230 GOSUB 300
240 NEXT I
250 RETURN
300 IF N=0 THEN LET B$=B$+" "
310 IF N=1 THEN LET B$=B$+"GA"
320 IF N=2 THEN LET B$=B$+"BU"
330 IF N=3 THEN LET B$=B$+"ZO"
340 IF N=4 THEN LET B$=B$+"MEU"
350 RETURN
500 REM TRADUCE B$(SHADOK) A
CASTELLANO (A$)
510 REM
520 LET I=1
530 LET A$=""
540 GOSUB 700
550 LET N1=N
560 GOSUB 700

```

```

570 LET N2=N
580 GOSUB 700
590 LET N3=N
600 LET A$=A$+CHR$ (N3+5*N2+25*
N1)
610 IF I>LEN B$ THEN RETURN
620 GOTO 540
700 IF I>LEN B$ THEN GOTO 900
710 LET N=0
720 IF B$(I)=" " THEN LET N=0
730 IF B$(I)="G" THEN LET N=1
740 IF B$(I)="B" THEN LET N=2
750 IF B$(I)="Z" THEN LET N=3
760 IF B$(I)="M" THEN LET N=4
770 IF N=0 THEN LET I=I+1
780 IF N=1 OR N=2 OR N=3 THEN L
ET I=I+2
790 IF N=4 THEN LET I=I+3
800 IF N=5 THEN GOTO 900
810 RETURN
900 PRINT "ERROR EN SU TEXTO SH
ADOK"
910 PRINT "HACER NEWLINE"
920 INPUT A$
930 RUN
940 REM
950 REM
1000 PRINT "INTRODUCIR EL TEXTO
CASTELLANO"
1010 INPUT A$
1020 LET B$=""
1030 CLS
1040 PRINT "TRADUCCION EN CURSO"
1050 GOSUB 100
1060 GOTO 2000
1500 REM
1510 PRINT "INTRODUCIR EL TEXTO
SHADOK"
1520 INPUT B$
1530 LET A$=""
1540 CLS
1550 PRINT "TRADUCCION EN CURSO"
1560 GOSUB 500
2000 REM
2010 CLS
2020 PRINT " ";B$;" "
2030 PRINT
2040 PRINT " (";A$;" ) "
2050 INPUT R$
2060 IF R$<>"C" THEN RUN
2070 LPRINT " ";B$;" "
2080 LPRINT
2090 LPRINT " (";A$;" ) "
2100 LPRINT
2110 LPRINT
2120 LPRINT
2130 RUN

```

```

*GABUMEUBUGAZOGAZOBUBU GABU BUBU
GAGA GAZOGAGAMEUGAGABUZOBUGAGA
*

```

(BUENOS DIAS)

```

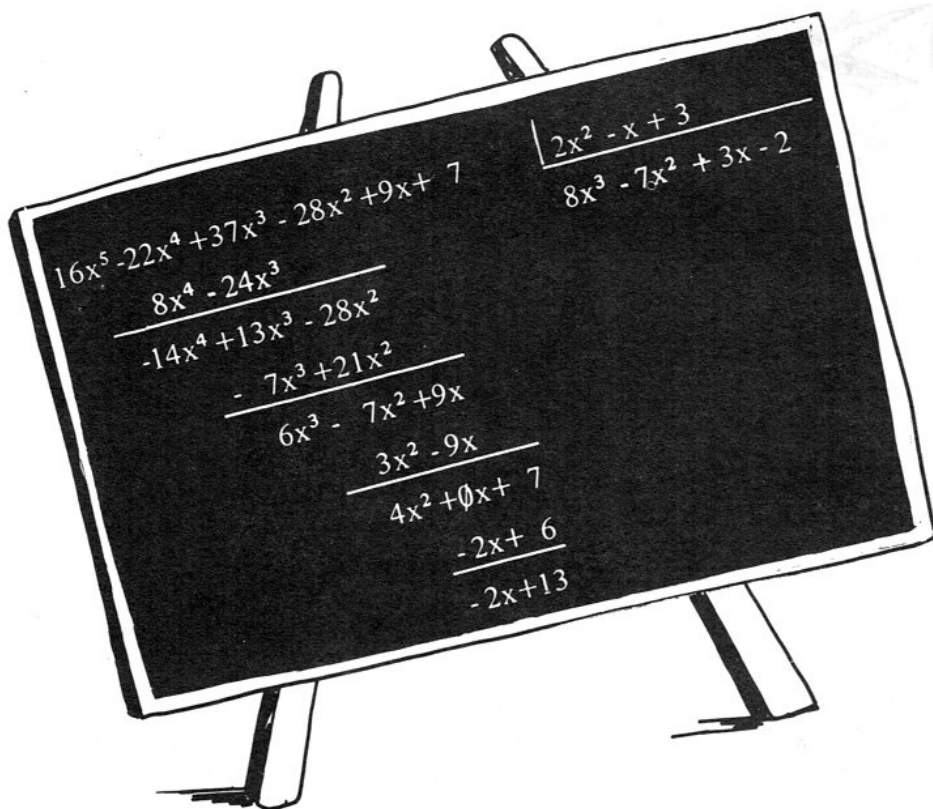
*GAZOBUGAMEUMEU BU BUBUGA GAZO
GAGAZOBUBU GAGABUZOGAGAZABU BUBU
GA BU ZOGAZOBUBUGA BUGAGAGA Z
OBU GAGABUZOGAMEUMEU BU BUBUGA
GA GAMEUGABU GAGAMEUGAGAZO GA
BUGAGABUZO GABUZOGAMEUMEU GA
MEUMEUGAZOBUBU GAGAZOEBUGAZOGA
BUGAGAMEUBUGAZOBU BUGAGAMEU
GABUZOGAZOGABU BUGAMEUZO*

```

(EL ORDENADOR PERSONAL OS INICIA
AL LENGUAJE SHADOK.)

DIVISION DE POLINOMIOS

José Manuel Rodríguez Prolongo y
José Antonio Mañas Valle.



— La tabla A será la encargada de guardar el dividendo y también las operaciones de suma que se realicen bajo él, así como el resto. Por tanto, esta tabla deberá de tener tantas columnas como sea el valor de M, y tantas filas como indique la siguiente expresión $2x(M-N) + 2$. A esta expresión se llega tras estudiar detenidamente muchas divisiones, entonces se observa que la longitud de la división dependerá de dicha expresión. Ya tenemos, pues, dimensionada la tabla A (650), y tras limpiarla (700 a (1100) pasamos a hacer lo mismo con la tabla B.

— En la tabla B se almacenarán el divisor y el cociente, por tanto, tan sólo necesitará dos filas (la 0 y la 1). El número de columnas variará según la longitud del cociente o del divisor, en cualquier caso siempre serán menor que M, por tanto, este puede ser su número de columnas. El dimensionado y limpieza de esta tabla se realiza en las líneas 1200 a 1700.

Introducción de los datos.

En primer lugar se introducen los valores de M y N, que han de ser naturales, y que en el caso de ser introducidos datos erróneos (caracteres alfabéticos, números negativos o números con decimales), éstos serán rechazados volviendo a pedir una nueva intro-

El programa realiza la operación división entre dos polinomios del tipo $A_0X^n + A_1X^{n-1} \dots A_{(N-1)}X + A_N$, dando como resultado un cociente y un resto, ambos también de la forma anterior. El programa ha sido realizado para el ordenador ATARI 800.

Es normal y no sorprende a nadie el hecho de que un ordenador sea capaz de tratar una serie de datos numéricos dando como resultado otros datos también numéricos. No obstante, choca un poco ver a un ordenador operar, no solo con números, sino con expresiones completas, como pueden ser, por ejemplo, los polinomios de la forma indicada.

El programa, cuyo listado se acompaña, admite muchas mejoras, pero puede servir de base para futuros desarrollos.

Planteamiento.

Partimos del estudio de una división de polinomios en su forma convencional. (Ver recuadro).

Llamaremos M al mayor exponente del dividendo, y N al del divisor, siendo M mayor o igual a N.

Para que el ordenador pueda realizar las operaciones tal y como se efectúan en el papel, hemos de reservar el espacio suficiente en unas tablas que tendrán una dimensión determinada.

$$\begin{array}{r}
 16x^5 - 22x^4 + 37x^3 - 28x^2 + 9x + 7 \\
 \underline{8x^4 - 24x^3} \\
 -14x^4 + 13x^3 - 28x^2 \\
 \underline{-7x^3 + 21x^2} \\
 6x^3 - 7x^2 + 9x \\
 \underline{3x^2 - 9x} \\
 -4x^2 + 0x + 7 \\
 \underline{-2x + 6} \\
 -2x + 13
 \end{array}
 \quad
 \begin{array}{r}
 2x^2 - x + 3 \\
 \underline{8x^3 - 7x^2 + 3x - 2}
 \end{array}$$


```

100 REM PROGRAMA PARA DIVIDIR POLINOMIOS
110 REM PROGRAMADORES: J.M.RODRIGUEZ PROLONGO J.A.MAÑAS VALLE 15-4-82"
120 DIM M$(5)
140 DIM N$(5)
160 DIM PA$(20)
180 DIM PE$(20)
200 PRINT "PROGRAMA PARA DIVIDIR POLINOMIOS TIPO:"
250 PRINT "A0X^M+...+(M-1)X+AM/BOX^N+B1X^(N-1)+...+B(N-1)X+BN"
300 PRINT "SIENDO M=N O M>N, Y AMBOS Nds NATURALES"
320 REM -----
340 REM VALOR DEL MAYOR EXPONENTE (M)
410 PRINT "INTRODUZCA EL VALOR DE M"
420 INPUT M$
430 LET JO=ASC(M$)
440 IF JO>57 OR JO<48 THEN 470
450 LET M=VAL(M$)
460 GOTO 480
470 PRINT "M=";M$;" ERROR.M SOLO ADMITE VALORES DE NUMEROS NATURALES";GOTO 410
480 LET VA=M-INT(M)
490 IF M<0 OR VA<>0 THEN PRINT "M=";M$;" ERROR.M HA DE SER UN NUMERO NATURAL";GOTO 410
500 REM VALOR DEL MENOR EXPONENTE (N)
510 PRINT "INTRODUZCA EL VALOR DE N"
520 INPUT N$
530 LET JI=ASC(N$)
540 IF JI>57 OR JI<48 THEN 570
550 LET N=VAL(N$)
560 GOTO 580
570 PRINT "N=";N$;" N=";N$;" ERROR.N SOLO ADMITE VALORES DE NUMEROS NATURALES";GOTO 510
580 LET VE=N-INT(N)
590 IF N<0 OR VE<>0 THEN PRINT "N=";N$;" N=";N$;" ERROR.N HA DE SER UN NUMERO NATURAL";GOTO 510
600 IF M<N THEN PRINT "M=";M$;" N=";N$;" ERROR. M HA DE SER MAYOR O IGUAL QUE N";GOTO 410
620 REM -----
640 REM DIMENSIONADO Y LIMPIEZA DE TABLAS
650 DIM A(2*(M-N)+2,M)
700 FOR U=0 TO 2*(M-N)+2
800 FOR V=0 TO M
900 LET A(U,V)=0
1000 NEXT V
1100 NEXT U
1200 DIM B(1,M)
1300 FOR X=0 TO 1
1400 FOR Y=0 TO M
1500 LET B(X,Y)=0
1600 NEXT Y
1700 NEXT X
1750 REM -----
1770 REM ASIGNACION DE VALORES A LAS CONSTANTES
1800 PRINT "INTRODUZCA DE A0 A AM"
1900 FOR K=0 TO M
1910 PRINT "A";K;
1920 INPUT PA$
1940 LET TI=ASC(PA$)
1960 IF TI>57 OR TI<45 THEN 2020
1970 IF TI=47 THEN 2020
1990 GOTO 2030
2020 PRINT "ERROR.A SOLO ADMITE VALORES NUMERICOS";GOTO 1910
2030 LET PA=VAL(PA$)
2050 LET A(0,K)=PA
2100 NEXT K
2200 PRINT "INTRODUZCA DE B0 A BN"
2300 FOR R=0 TO N
2310 PRINT "B";R;
2320 INPUT PE$
2340 LET TO=ASC(PE$)
2360 IF TO>57 OR TO<45 THEN 2420
2370 IF TO=47 THEN 2420
2380 GOTO 2430
2420 PRINT "ERROR.B SOLO ADMITE VALORES NUMERICOS";GOTO 2310
2430 LET PE=VAL(PE$)
2450 LET B(0,R)=PE
2500 NEXT R
2550 REM -----
2600 REM LAZO DE OPERACIONES
2700 FOR H=0 TO M-N
2800 LET B(1,H)=A(2*(H),H)/B(0,0)
2900 FOR C=H+1 TO N+H
3000 LET A(2*(H+1),C)=B(1,H)*B(0,C-H)
3100 NEXT C
3200 FOR D=H+1 TO N+H
3300 LET A(2*(H+1),D)=A(2*(H),D)+A(1+2*(H),D)
3400 NEXT D
3500 IF H=M-N THEN 3700
3600 LET A(2*(H+1),H+N+1)=A(0,H+N+1)
3700 NEXT H
3750 REM -----
3770 REM LAZO DE IMPRESION DEL DIVIDENDO
3800 PRINT "DIVIDENDO: ";
3900 LET D=M
4000 FOR I=0 TO M
4100 IF A(0,I)<0 THEN 4700
4200 IF D=0 THEN 4500
4250 IF D=1 THEN 4650
4300 PRINT "+";INT(A(0,I)*100+0.5)/100;"X";D;
4400 GOTO 5100
4500 PRINT "+";INT(A(0,I)*100+0.5)/100
4600 GOTO 5100
4650 PRINT "+";INT(A(0,I)*100+0.5)/100;"X";
4670 GOTO 5100
4700 IF D=0 THEN 5000
4750 IF D=1 THEN 5050
4800 PRINT INT(A(0,I)*100+0.5)/100;"X";D;
4900 GOTO 5100
5000 PRINT INT(A(0,I)*100+0.5)/100
5030 GOTO 5100
5050 PRINT INT(A(0,I)*100+0.5)/100;"X";
5100 LET D=D-1
5200 NEXT I
5250 REM -----
5270 REM LAZO DE IMPRESION DEL DIVISOR
5300 PRINT "DIVISOR: ";
5400 LET E=N
5500 FOR F=0 TO N
5600 IF B(0,F)<0 THEN 6200
5700 IF E=0 THEN 6000
5750 IF E=1 THEN 6150
5800 PRINT INT(B(0,F)*100+0.5)/100;"X";E;
5900 GOTO 6600
6000 PRINT "+";INT(B(0,F)*100+0.5)/100
6100 GOTO 6600
6150 PRINT "+";INT(B(0,F)*100+0.5)/100;"X";
6170 GOTO 6600
6200 IF E=0 THEN 6500
6250 IF E=1 THEN 6550
6300 PRINT INT(B(0,F)*100+0.5)/100;"X";E;
6400 GOTO 6600
6500 PRINT INT(B(0,F)*100+0.5)/100
6530 GOTO 6600
6550 PRINT INT(B(0,F)*100+0.5)/100;"X";
6600 LET E=E-1
6700 NEXT F
6750 REM -----
6770 REM LAZO DE IMPRESION DEL COCIENTE
6800 PRINT "COCIENTE: ";
6900 LET G=M-N
7000 FOR H=0 TO M-N
7100 IF B(1,H)<0 THEN 7700
7200 IF G=0 THEN 7500
7250 IF G=1 THEN 7650
7300 PRINT "+";INT(B(1,H)*100+0.5)/100;"X";G;
7400 GOTO 8100

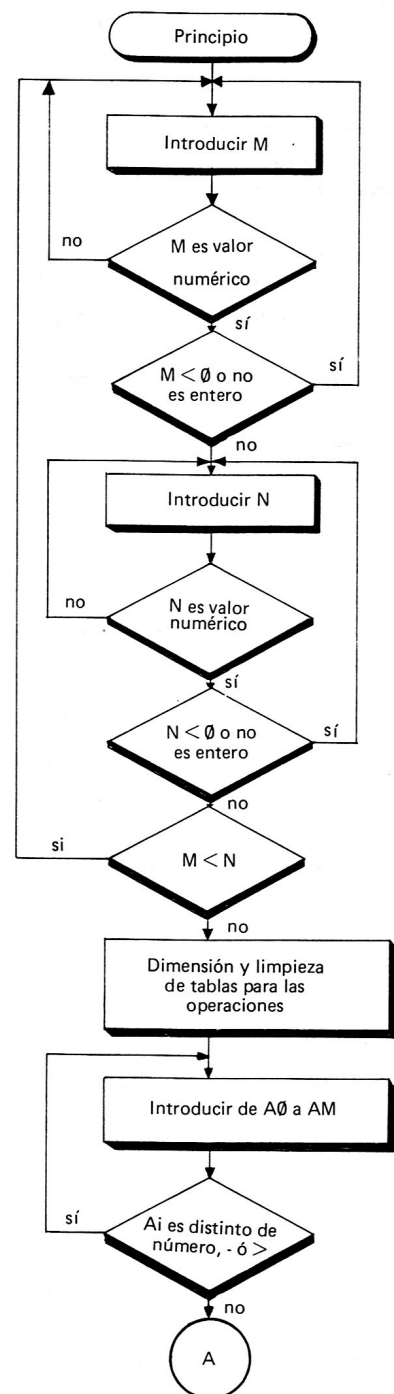
```

```

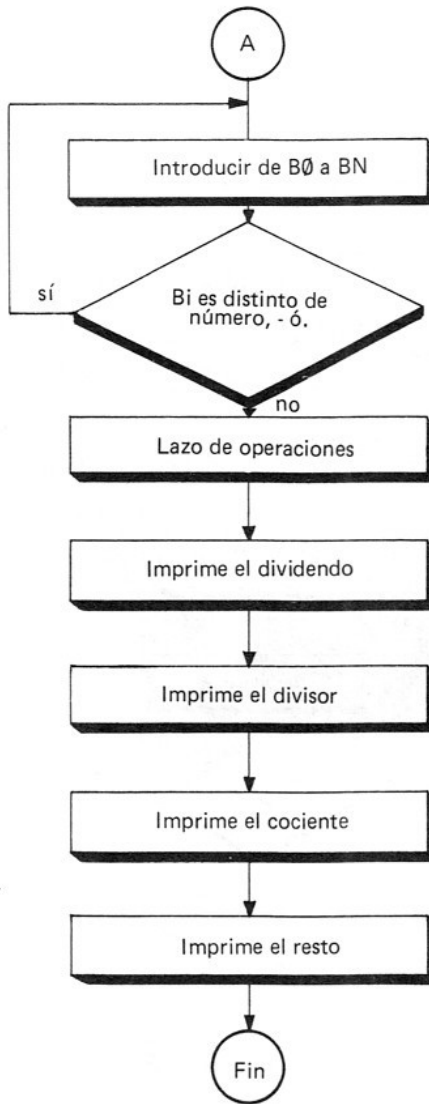
7500 PRINT "+";INT(B(1,H)*100+0.5)/100
7600 GOTO 8100
7650 PRINT "+";INT(B(1,H)*100+0.5)/100;"X";
7670 GOTO 8100
7700 IF G=0 THEN 8000
7750 IF G=1 THEN 8050
7800 PRINT INT(B(1,H)*100+0.5)/100;"X";G;
7900 GOTO 8100
8000 PRINT INT(B(1,H)*100+0.5)/100
8030 GOTO 8100
8050 PRINT INT(B(1,H)*100+0.5)/100;"X";
8100 LET G=G-1
8200 NEXT H
8250 REM -----
8270 REM LAZO DE IMPRESION DEL RESTO
8300 PRINT "RESTO: ";
8400 LET J=N-1
8500 FOR K=M-N+1 TO M
8600 IF A(2*(M-N)+2,K)<0 THEN 9200
8700 IF J=0 THEN 9000
8750 IF J=1 THEN 9150
8800 PRINT "+";INT(A(2*(M-N)+2,K)*100+0.5)/100;"X";J;
8900 GOTO 9600
9000 PRINT "+";INT(A(2*(M-N)+2,K)*100+0.5)/100
9100 GOTO 9600
9150 PRINT "+";INT(A(2*(M-N)+2,K)*100+0.5)/100;"X";
9170 GOTO 9600
9200 IF J=0 THEN 9500
9250 IF J=1 THEN 9550
9300 PRINT INT(A(2*(M-N)+2,K)*100+0.5)/100;"X";J;
9400 GOTO 9600
9500 PRINT INT(A(2*(M-N)+2,K)*100+0.5)/100
9530 GOTO 9600
9550 PRINT INT(A(2*(M-N)+2,K)*100+0.5)/100;"X";
9600 LET J=J-1
9700 NEXT K
9800 END

```

Algoritmo del programa de división de polinomios.



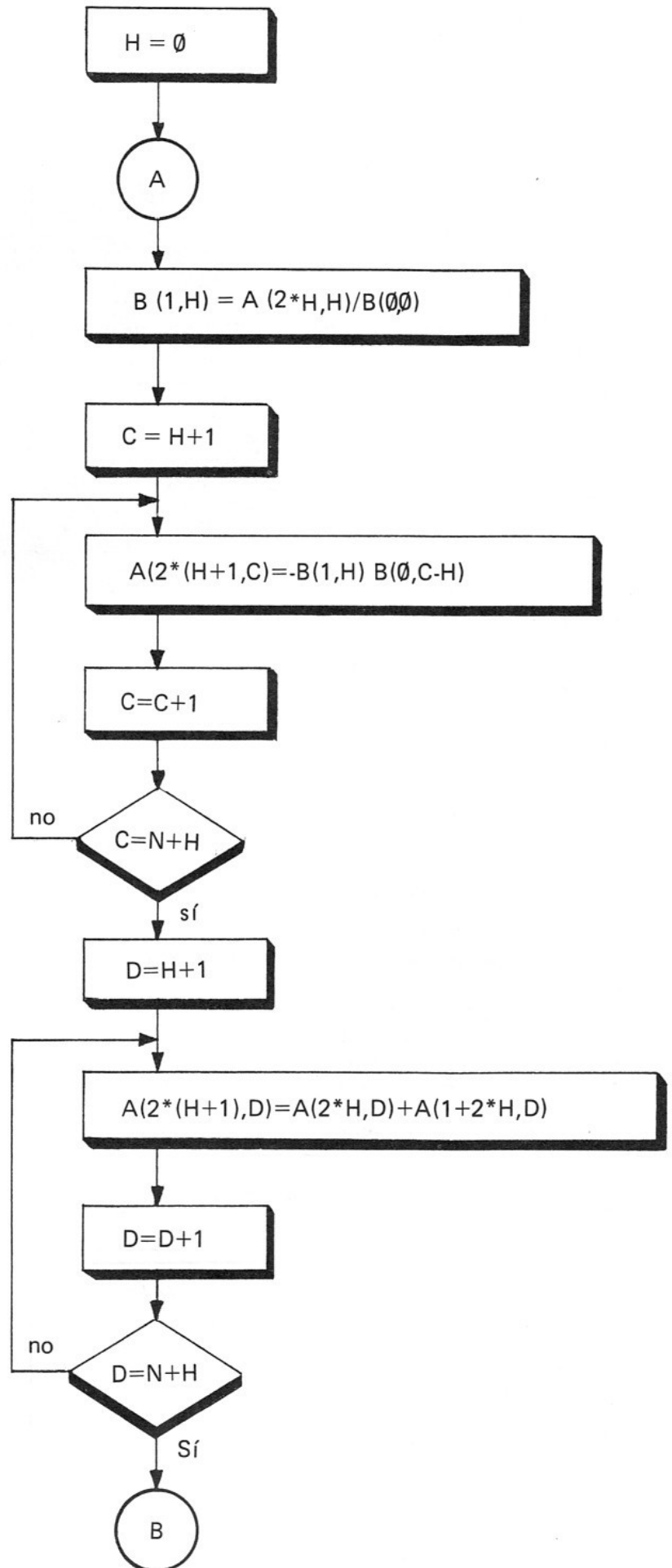
Organigrama del lazo de operaciones.

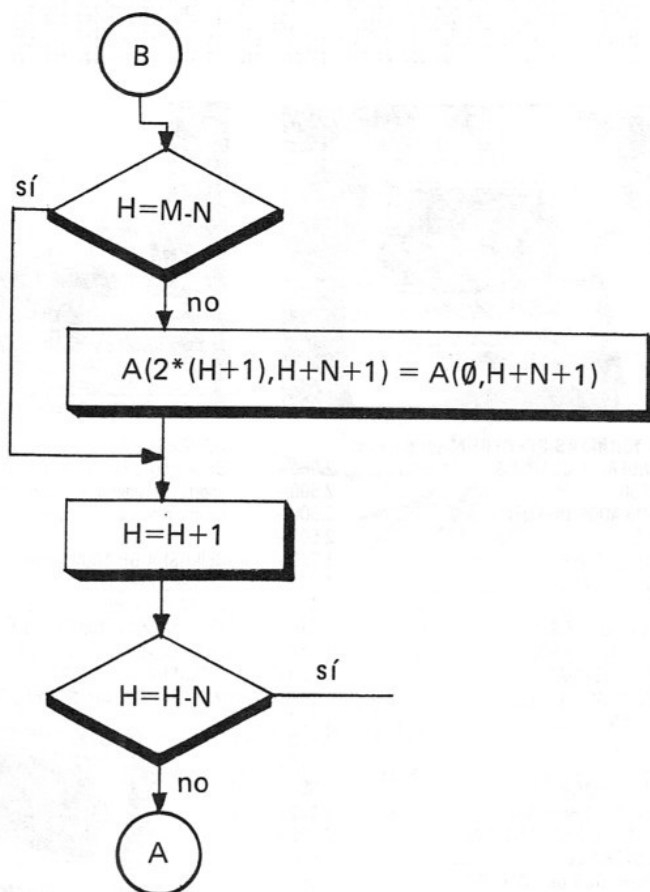


ducción del dato erróneo. Igualmente, en el caso de que N sea mayor que M se repetirá la entrada de los datos (410 a 600).

Como fácilmente puede imaginarse, el ordenador tan sólo va a operar con las constantes de las expresiones, o sea, que sólo términos de la expresión y no sus exponentes, excepto como ya hemos visto antes, al principio hay que introducir M y N para poder dimensionar.

— Para introducir el dividendo hacemos uso de un lazo que se repetirá de 0 a M veces, dentro del cual hay una INPUT (1920) y además va introduciendo cada dato en su lugar correspondiente de la tabla A(2050). Al mismo





tiempo, dentro de este lazo hemos intercalado sentencias IF... THEN, para que en el caso de que se introduzcan datos no válidos (caracteres no numéricos distintos de menos o de la coma decimal) vuelva a repetir el ciclo en que se encuentra.

— La introducción del divisor se realiza de la misma forma en la fila 0 de la tabla B (2200 a 2500).

Lazo de operaciones.

Es esta parte del programa la más importante, y quizás por eso la de más difícil comprensión (2600 a 3700).

En el ejemplo anteriormente expuesto podemos observar cómo la operación de dividir polinomios consiste en realizar consecutivamente las siguientes operaciones: división de números, multiplicación de números, cambio de signo y suma de números. Este proceso se realizará M-N veces, ya que cada vez que se realice, el exponente mayor del dividendo disminuirá en un grado

hasta que éste sea menor a N, en cuyo caso acabará la división, por lo tanto, el lazo principal de operaciones se realizará tantas veces como indique la sentencia 2700.

Dentro de este lazo que podríamos llamar general, existen otros dos. El primero consiste en multiplicar cada elemento del cociente por cada uno de los del divisor y cambiarlos de signo, llevando el resultado a la posición correspondiente de la tabla A (2900 a 3100); se realiza tantas veces como indique N. El segundo lazo consiste en sumar los valores que obtengamos en el lazo anterior con sus inmediatos superiores en la tabla A; este lazo también se realizará N veces (3200 a 3400).

Tras realizar estos dos lazos y siempre dentro del lazo general, es necesario bajar el siguiente término tal y como se hace en la división de polinomios convencional (3600), este paso no se realiza en el último lazo, ya que no quedan más términos en el dividendo (3500).

Como hemos dicho antes, estos dos lazos se realizan dentro de uno general, y en cada vuelta de éste los distintos valores que se obtengan en los dos lazos secundarios se situarán en distintas posiciones dentro de las tablas, pero siempre siguiendo la misma disposición que en las divisiones de polinomios convencionales.

Debe quedar claro que en el lazo de operaciones se ha jugado tan sólo con los valores constantes de cada factor, y que su exponente sólo se ha tenido en cuenta para su distribución en las tablas.

Estas operaciones están claramente reflejadas en el Organigrama del Lazo de Operaciones.

Impresión del resultado.

El programa realiza la impresión en pantalla del dividendo, divisor, cociente y resto.

— Dividendo: basta con imprimir los valores que hay en la fila 0 de la tabla A, dotando a cada número de su correspondiente X y del exponente de cada X, y también teniendo en cuenta el signo (4100). Se realiza, pues, la impresión con un lazo que se repetirá M veces, en la cual habrá una variable (0) que irá disminuyendo su valor en cada vuelta y que representará el valor del exponente de X. Hemos dotado a este lazo de algunas IF... THEN, con objeto de eliminar el exponente de X en el caso de que éste sea 1, o de eliminar a X en el caso de que su exponente sea 0 (3800 a 5200).

— Divisor: se realiza de la misma forma, pero imprimiendo la fila 0 de la tabla B, y su lazo se repetirá N veces. (5300 a 6700).

— Cociente: igualmente con la fila 1 de la tabla B y se hará M-N veces (6800 a 8200).

— Resto: igualmente con la última fila de la tabla A, es decir, con la 2x (M-N) + 2, repitiendo la secuencia N-1 vez (8300 a 9700).

Todos los resultados van redondeados hasta 2 cifras decimales.

Lo aquí expuesto queda reflejado de una forma muy general en el algoritmo de este programa. □

VENEA A GUBERNALU

Productos garantizados por los importadores oficiales.
Evítese sorpresas desagradables.

commodore

COMMODORE 64	64.900,-
CASSETTE (grabador-reproductor)	12.000,-
UNIDAD DE DISCO (170K)	95.000,-
PADDLES (PAREJA)	2.500,-

ZX Spectrum

SPECTRUM 16K	31.900,-
SPECTRUM 48K	41.000,-
SPECTRUM (ampliado 48K)	39.900,-
AMPLIACION DE 16K A 48K (montada)	8.000,-
ZX INTERFACE 1	17.500,-
ZX MICRODRIVE	17.500,-
ZX INTERFACE 2	7.900,-

ORIC ATMOS, SPECTRAVIDEO SV 318, SPECTRAVIDEO SV 328, UNITRON, MULTITECH, APPLE, etc...



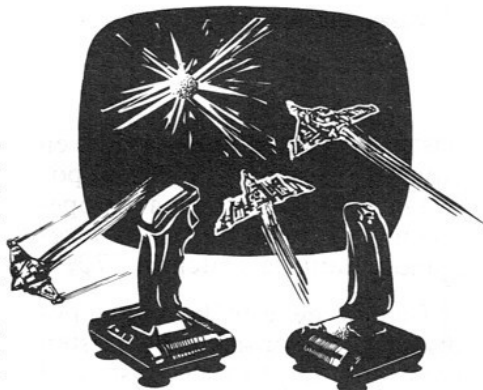
PROGRAMAS COMMODORE

CONTABILIDAD C-64 (P.G.C.)	24.550,-
MAGIC DESK (cart.)	16.000,-
LE MANS C64 (cart.)	4.500,-
MUSIC MACHINE (cart.)	4.500,-
SEA WOLF (cart.)	4.500,-
LENGUAJE MAQUINA C64 (cart.)	10.000,-
MASTER 64 (disco)	25.000,-
CURSILLOS PROGRAMACION (basic, cod. máquina, sprites) cas.	2.600,-
FALCON PATROL (cas.)	2.500,-
PIT STOP coches (cas.)	2.500,-
MANIC MINER (cas.)	3.000,-
NAVATRON (cas.)	3.000,-
SUPER DOGFIGHT (cas.)	2.000,-
ELECTRO LASER (cas.)	2.000,-
HUNDBACH (cas.)	3.000,-
PIPELINE el fontanero (cas.)	3.000,-
REVENGE OF MUTANT CAMELS (cas.)	3.000,-
GRAND MASTER ajedrez (cas.)	3.000,-



PROGRAMAS SPECTRUM (originales)

BANDERA A CUADROS	2.000,-
TUTOR	2.500,-
SIMULADOR DE VUELO	2.500,-
VU 3D	2.500,-
MANIC MINER	1.700,-
JET SET WILLY	1.700,-
ANDROIDE UNO	1.485,-
ANDROIDE DOS	1.580,-
ESCALADOR LOCO	1.580,-
LUNAR JETMAN	1.800,-
ATIC ATAC	1.800,-
ALCHEMIST	1.800,-
JETPAC	1.800,-
PPSSST	1.800,-
THE PYRAMID	1.800,-
ROMMEL'S REVENGE	1.800,-
CONTABILIDAD PERSONAL	2.320,-
MASTER FILE	2.500,-
GENERADOR DE ROTULOS	2.200,-

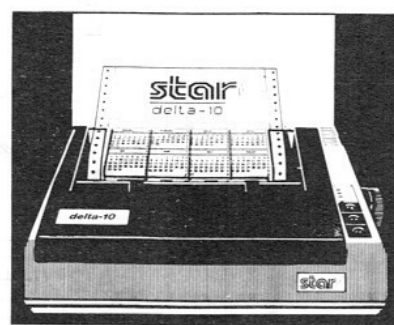


NOVEDADES

CURSO INTR. BASIC II C-64	3.000,-
SIMON'S BASIC C-64	16.000,-
JOY STICK CONTROL REMOTO (par.)	6.900,-
SIM. VUELO C-64 (disco)	7.000,-
FUTBOL + JUMPMAN (disco)	5.000,-
FUNDAS COMMODORE	1.100,-
LAPIZ OPTICO	9.900,-
IMPRESORA RITEMAN Folio, portátil	81.000,-
IMPRESORA SEIKO GP-50 SPECTRUM	26.900,-
PAPEL GP-50	150,-

OFERTAS

	ANTES	OFERTA
JOY STICK QUICKSHOOT II	4.200,-	2.500,-
DISCOS WABASH 5 1/4 (10 U.)	5.080,-	3.840,-
DISCOS NASHUA FD 2D 8" 10 u.	10.240,-	7.050,-
MONITOR FONTEC 9" f.v.	28.500,-	25.556,-



IMPRESORAS

Se la instalamos a su ordenador, ahorro considerable en cables y conectores. Le iniciamos en su manejo.

SEIKOSHA GP 100VC (commodore)	49.500,-
COMMODORE MPS 801	59.900,-
ADMATE DP 80	59.500,-
STAR GEMINI 10X 120 c.p.s.	79.500,-
STAR GEMINI 15X	115.000,-
SEIKOSHA GP 700 color	98.500,-
INTERFACE STAR-COMMODORE	10.000,-



ASESORAMIENTO EN LA MECANIZACION DE PEQUEÑAS EMPRESAS Y APLICACIONES ESPECIFICAS.

SOLUCIONES desde 223.155,-
(Unidad central 64K, lector de discos e impresora 80 c.p.s.)

PROGRAMAS: Procesado de textos, Contabilidad, Mailing (etiquetas direcciones), Fichero clientes, Hojas electrónicas (presupuestos, previsiones de ventas)...

PERIFERICOS Y ACCESORIOS

INTERFACE APPLE/IMPRESORA GRAFICA	16.390,-
DISCOS MAXELL 5 1/4"	508,-
DISCOS DATALIFE	586,-
PAPEL PAUTADO 3.000 h.	3.880,-
PAPEL BLANCO 2.700 h.	4.075,-
PAPEL PAUTADO GP 80 1.000 h	999,-
KIT LIMPIEZA FLOPPY 5 1/4	4.227,-
MONITOR FONTEC 12" f.v.	30.100,-
MONITOR MULTITECH (Base móvil)	38.000,-
CAJA ARCHIVO 50 DISCOS	3.975,-
CAJA ARCHIVO 25 DISCOS	6.220,-



GRANDES OFERTAS

GUBERNALU

ELECTRONICA, S.A.

Sepúlveda, 104 - Barcelona-15 - España

Tels. (93) 224 37 27 - 223 49 12 - 223 42 43

243 34 32. Telex 59.123 GBRN E

Choque elástico



Este programa permite simular en la pantalla del televisor, el choque elástico entre dos discos rígidos, en conformidad con las leyes de la Física. Aunque el programa se ha realizado para el «Dragón 32», se dan algunas indicaciones para su adaptación a otros miniordenadores.

Características

El presente programa tiene una finalidad fundamentalmente didáctica: facilitar el estudio teórico y experimental del choque elástico en dos dimensiones, ya que los choques entre bolas o entre discos no resultan fáciles de estudiar debido a los rozamientos y a las dificultades de medida. Este programa permite realizar y observar fácilmente una fiel simulación del choque elástico entre dos discos, haciendo posible la elección de los parámetros que caracterizan el choque (radio y masa de los discos y ángulo de incidencia) y ofreciendo tablas y esquemas que describen completamente las características del choque.

Al observarse el choque, puede por ejemplo, cronometrarse la duración del movimiento de los discos antes y después del choque, medirse sobre la pantalla del televisor las distancias recorridas por los discos y a partir de estos datos, calcularse la velocidad de

los discos antes y después del choque. (1)

Un poco de física

Consideremos el choque de un disco que se mueve con una velocidad y dirección dadas (proyectil), contra otro disco que se encuentra en reposo (blanco). Para determinar completamente la velocidad después del choque de cualquiera de los dos discos, hemos de conocer dos valores, el del módulo de la velocidad y el del ángulo que ésta forma con la horizontal, o los de sus componentes horizontal y vertical. Por lo tanto, para determinar completamente el movimiento de los dos discos después del choque, necesitamos conocer los valores de las cuatro variables que definen las velocidades de los dos discos.

En ausencia de rozamientos y considerando un choque perfectamente elástico en dos dimensiones, las leyes de la física nos indican que se conserva la energía cinética (nos aporta 1 ecuación) y el momento lineal o cantidad de movimiento ($m \cdot v$, producto de la masa por la velocidad) que nos aporta 2 ecuaciones (una por cada componente del momento lineal o de la velocidad).

Tenemos por lo tanto un sistema de 3 ecuaciones con cuatro variables cuya solución no será única sino que se obtendrá un conjunto de soluciones que dependen de un parámetro arbitrario. En nuestro caso este parámetro describe el modo en el que se efectúa el choque, es decir, la forma de los cuerpos, sus zonas de contacto y cómo se deformaron en el choque. En el caso de dos discos rígidos y homogéneos, este parámetro queda determinado por el hecho de que en el sistema de referencia de centro de masa (2), los movimientos del centro del proyectil antes y después del choque, son simétricos respecto de la recta que une los centros de los dos discos en el momento del choque. Esto nos permite calcular el ángulo que con la horizontal, forma la velocidad del proyectil después del choque y de esta forma queda perfectamente definido el movimiento de los discos después del choque.

Debido a dificultades matemáticas y experimentales, el estudio de los choques se limita usualmente al de un caso particular denominado choque frontal. Un choque entre dos discos puede considerarse frontal, cuando en el sistema de referencia en el que uno de ellos está en reposo, la trayectoria del centro del disco móvil tiende a pasar, en el momento del choque por el centro del disco en reposo. En el sistema de centro de masa los dos discos se moverían sobre la misma recta,

(1) Algunas consideraciones didácticas y otras sugerencias para su empleo en el aula, pueden consultarse en Christian WAGNER, «Experiencias mediante ordenador en Física y Química», Revista Patio Abierto nº 7, ICE de La Universidad de Cádiz.

(2) El sistema de referencia de centro de masa de los dos discos es un sistema de referencia en el que los centros de ambos discos se mueven siempre sobre rectas paralelas, en sentidos opuestos y de modo que el momento lineal de ambos discos sea el mismo.

antes y después del choque. Aunque en este programa el choque frontal es una de las muchas posibilidades, el programa ofrece la opción de elegir él la dirección del movimiento del proyectil de modo que el choque sea frontal.

Advirtamos que al limitarnos al movimiento en dos dimensiones, es preferible referirnos al movimiento de discos en lugar de al de bolas. De esta forma nuestro modelo se acerca más a la realidad ya que en el caso de los discos no hay que considerar el momento angular ni la energía de rotación, como usualmente se hace. Además, si las dos bolas no tienen igual tamaño, la proyección sobre el plano de la recta que une los centros de las bolas en el momento del choque, será menor que la suma de sus radios, lo que dificultaría su representación en la pantalla y haría perder claridad a la simulación.

Descripción del programa

Este programa ha sido realizado para el **Dragón 32** y utiliza **24,5K** de su memoria RAM. Sin embargo puede adaptarse a otros miniordenadores que tengan posibilidades gráficas similares aunque su capacidad de memoria sea menor, según iremos indicando.

El programa está estructurado en un bloque inicial y en nueve bloques que comienzan por una sentencia REM, como puede observarse en el listado.

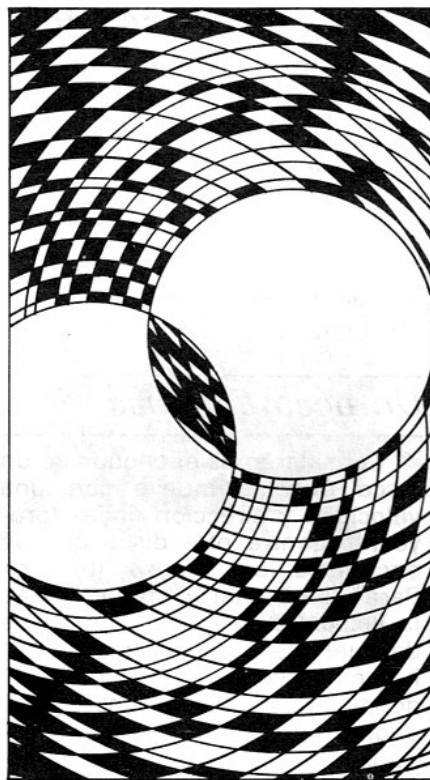
En el bloque inicial, se dimensionan dos matrices y se asignan valores a las constantes y variables que se necesitarán para obtener el movimiento de los discos. Al hacerlo de esta forma, se reduce el tiempo que tarda el ordenador en localizar dichas variables y constantes, aumentando la velocidad máxima con la que pueden moverse los discos sobre la pantalla. Si se desea incrementar la velocidad de los discos en un 35%, tecléese POKE 65495,0 antes de ejecutar el programa (3). En este caso disminuye ligeramente el ajuste de las velocidades. Para volver a la velocidad normal tecléese POKE 65494,0.

El bloque 1º sólo realiza el cartel presentador del programa y el bloque 2º da en pantalla algunas indicaciones para su utilización, por lo que pueden su-

primirse sin alterar el programa, siempre que se conserve la línea 56.

Al no poder mezclarse la alta resolución con texto de pantalla, el bloque 3º prepara algunos rótulos y letras que se utilizarán en alta resolución e incluye una subrutina que sitúa unos rótulos en la esquina izquierda de la pantalla (líneas 86 y 88).

En el bloque 4º se introducen los datos a través del teclado y del JOYSTICK izquierdo. Anotemos que a menos de un error de conexión de mi ordenador, en el manual están cambiadas las instrucciones del joystick derecho e izquierdo. Advirtamos también que hemos tenido que introducir la llamada JOYSTK(0) antes de JOYSTK(3) que era la que nos interesaba, pues de lo contrario dicha llamada no se efectuaba (línea 152).



Aunque la masa del proyectil se considera igual a la unidad, esto no quita generalidad al programa ya que la masa del blanco puede tomar cualquier valor, mayor o menor que uno.

El radio del proyectil es de 5 puntos mientras que el del blanco se elige entre 5 y 17 puntos. Aunque no parece interesante utilizar blancos mayores, su radio podría aumentarse hasta 50 puntos pues hay 7,5K de memoria disponible. Por el contrario si se modifica el programa de modo que los dos discos tengan igual tamaño, puede utilizarse la misma matriz para ambos, ahorrándose memoria.

Como los puntos que el ordenador dibuja sobre la pantalla no son cuadrados sino rectangulares, en la línea 102 se introduce un factor de escala FE y su inverso EF. FE se obtiene dividiendo la longitud de una línea vertical por la de otra horizontal de igual número de puntos.

En el bloque 5º se obtienen los diferentes valores que permiten que el movimiento de los discos se adecúe a los datos introducidos y a las leyes de la física.

Los bloques 6º y 7º realizan, respectivamente, el movimiento de los discos antes y después del choque. El movimiento se efectúa colocando sucesivamente en la pantalla, mediante el comando PUT, un círculo verde rodeado de un filo negro que permite borrar los puntos no utilizados del círculo anterior.

Para que el movimiento de los discos antes y después del choque se adecúe a las leyes de la física se ha dispuesto un bucle FOR NEXT de frenado en la línea 216. El valor máximo del índice del bucle, IR, se obtiene mediante las líneas 98, 100 y 204 y su cálculo se explicará posteriormente. En la línea 252 se ha introducido otro bucle de frenado que puede ser necesario al modificar o adaptar el programa. Aunque el índice máximo de este bucle, JR, es cero, la eliminación de esta línea perjudicaría la bondad del ajuste obtenido.

Hay que observar que aunque en cada choque se considera la velocidad del proyectil igual a la unidad, ésta varía de un choque a otro de modo que la realización del choque dure lo menos posible. Advirtamos finalmente que entre la primera y las posteriores veces que se ejecuta el programa hay una ligera variación en la velocidad de los discos debido a la variación de los valores de T5 y T6 (líneas 220 y 256) y originada seguramente por una distinta posición inicial del puntero.

El bloque 8º permite presentar en alta resolución un esquema del choque. El dibujo de los ángulos que las trayectorias forman con la horizontal, así como las letras que los denominan, parcialmente definidas en el bloque 3º, se realizan mediante una subrutina que comprende las líneas 304 a 350. La primera parte de la subrutina (líneas 304 a 312) calcula el valor del ángulo.

El bloque 9º y último comienza presentando un menú de opciones. Entre ellas se encuentra la posibilidad de guardar en memoria el esquema del choque para compararlo con el que se obtenga

(3) Sugerencia de Colin CARTER en *Enter the Dragon*. Melbourne House, California, 1983.

CHOQUE ELASTICO

PROGRAMA ORIGINAL DE CHRISTIAN WAGNER

```

10 CLEAR 600 PCLERS
12 0=0 Y=0 YB=0 YB=0 YB=184 S=5 XL=254 VL=191 X1=0 Y1=0 X2=150 Y2=100
13 X3=0 Y3=0 X4=0 Y4=0 X5=0 Y5=0 R4=0 D=0 T1=0 K=0 C1=0 C2=0 C3=0 C4=0 E=0 E1=7 IL
14 0=0 YB=0 YB=1 DIM P(5) T5=0 T6=0 DIM RC(12) DIM BC(3)
14 REM 1 PRESENTACION DEL PROGRAMA *****
15 CLS
16 FOR I=0 TO 3
17 SET (4+1,3) SET (4+1,3) SET (4+1,7,3) SET (10+1,5,3)
18 SET (18+1,3,3) SET (21+4+1,3) SET (17,3+1,3) SET (17+1,7,3)
19 SET (25+1,3,3) SET (28+4+1,3) SET (25+1,7,3) SET (26+1,5,3) SET (26,6,3)
20 SET (32+1,7,3) SET (35+3+1,3) SET (38+1,3,3) SET (38+1,7,3) SET (38+4+1,3) SET (39,
5,3) SET (40,5,3)
21 SET (14+1,1,3) SET (14,12+1,2) SET (14+1,15,2) SET (15,13,3) SET (16,15,3)
22 SET (20,11+1,3) SET (20+1,15,3) SET (27+1,11,3) SET (27+1,13,3) SET (30,12+1,2)
23 SET (33+1,11,3) SET (33+1,13,3) SET (33+1,15,3) SET (35,12,3) SET (35,14,3)
24 SET (41,12+1,3) SET (49+1,11,3) SET (49+1,12+1,3) SET (49+1,15,3)
25 SET (55+1,11,3) SET (59,11+1,3) SET (55,12+1,3) SET (56+1,15,3)
26 NEXT I
27 FOR I=0 TO 4
28 SET (10,3+1,3) SET (14,3+1,3) SET (24,3+1,3) SET (31+1,3,3) SET (26,11+1,3) SET
39+1,11,3) SET (46,11+1,3)
40 NEXT I
41 PRINT 0448,"PROGRAMA DIDACTICO, ORIGINAL DE".
42 PRINT 0490," CHRISTIAN WAGNER"
43 FOR I=1 TO 500 NEXT I
44 REM 2 INDICACIONES *****
45 CLS
46 V=CHR$(159)+CHR$(159) JS=K+S GS=I+J+K+S
47 PRINT0," SIMULACION DEL CHOQUE DE UN PROYECTIL CONTRA UN BLANCO FIJO."
48 PRINT096,GS+"INDICACIONES"+GS:
49 PRINT0160,"DESPUES DE CADA INTERROGACION INTRODUCIR EL VALOR O RESPUESTA PE
DIDO, LUEGO PULSA ENTERA."
50 PRINT0288,"APUNTE MOVRIENDO VERTICALMENTE EL MANDO,DISPARE PULSANDO EL BOTON,TA
RA UN POCO ANTES DE COMENZAR."
66 PRINT0416,"PULSAR C PARA CONTINUAR AHORA Y DESPUES DE UN GRAFICO O TABLA."
68 H$=INKEY$
70 IF INKEY$<>"C" THEN 70
74 REM 3 LETRAS Y ROTULOS EN ALTA RESOLUCION *****
75 C$="L5,D9,R5,BP5,U9,B04,F5,B04,D9,BP5,U9,R5,D9,L5,BP10,U9,R5,D9,L5,BP2,U3,R2
,B05,BP5,D9,R5,U9,BP10,L5,D4,NK4,D5,R5"
76 H$="R5,D9,L5,U9,BL5,D9,BL5,U9,B02,F5"
80 F$="R4,BP1,B04,L5,D9,BP10,U9,BP5,D4,L5,F5,BP5,U9,R5,D9,L5,BP10,U9,B02,F5,B02,
U9,R5,BL5,D9,BP5,U9,R5,D9,BP10,L5,U9,BL5,B04,L5"82 GOTO 90
84 H$="BPB1,5"+H$ NR$="BM12,25"+F$
86 IF LY=136 THEN GS$="BM22,5"+C$ ELSE D$="BM10,5"+C$ DRAWNS
88 DRAWNS DRAWNS RETURN
90 H$="BP9,B01,U2,E1,U1,E1,U1,B012,D1,H3,U4,H2,U1,H4,L3,G2,D1,G1,D5,F1,D1,F2,R3,
E3,U1,U1,E1"
92 B$="B06,BL2,H1,U1,E4,P1,F1,R1,D1,G1,D1,G1,L5,F5,F3,G1,G1,L1,G1,L4"
94 C$="BP3,BU2,U3,U1,E1,P1,R1,F1,R1,F1,D1,G1,D1,G1,L1,G1,L1,L1,L1,H1
,U1,U1,U2,R5"
96 REM 4 CONSTANTES Y ENTRADA DE DATOS *****
98 DATA 167,05,172,46,176,58,183,50,189,67,197,86,206,45,215,86,222,61,233,75,24
5,12,255,57,270,91
100 FOR I=0 TO 12 READ RP(1) NEXT I
102 P1=ATN(K/4)+FE/5202 EPI=1,8667
106 CLS1:PRINT09,"I$="+ENTRADA DE DATOS"+J$+I$
108 PRINT064,"LA MASA,LA VELOCIDAD Y EL RADIO DEL PROYECTIL SE CONSIDERAN
IGUALES A LA UNIDAD."
110 PRINT0197,"DAR EL RADIO DEL BLANCO (ENTRE 1 Y 3.5 UNIDADES)"
112 INPUT R
114 PRINT0292,"DAR LA MASA DEL BLANCO"
116 PRINT0354,"SI SE LE DA EL VALOR CERO,EL PROGRAMA LO CALCULA,SUPONIENDO
0
VE EL BLANCO Y EL PROYECTIL TIENEN IGUAL DENSIDAD"
118 INPUT M
120 R$=INT(R*5) R3=RB+5 E=RB+1 R4=PB+R3
122 IF M=0 THEN M=RP
124 MM=1/(M+1)
126 CLS2
128 PRINT0192," ¿DESEA QUE LA COMPUTADORA LE ASEGURE UN CHOQUE FRONTAL?
130 INPUT CF$
132 X=0 Y=0 YB=X2 YB=Y2
134 PMODE4,1 SCREEN,0 PCLS
136 CIRCLE(X,Y),5.1 PRINT(X,Y),1.1
138 GET(X1,Y1)=X+E1 Y=E1 Y+Y1,P,IG
140 CIRCLE(XB,YB),R,B,1 Y=INT(XB,YB),1,1
142 GET(XB,YB)=X+RB Y=YB+YB+E,Y,B,G
144 PUT(X1,Y1)=X+E1 Y=E1 Y+Y1,P,PSET
146 A=36
148 LY=118/A/2
150 LINE(X,Y)-(90,LY),PSET
152 C=JOYSTK(X) IF CF$="SI" THEN A=36 ELSE A=JOYSTK(X)
154 LINE(X,Y)-(90,LY),PSET
156 C=PEEK(52200) IF C=128 AND C/255 THEN 148
158 REM 5 PREPARACION DE LOS MOVIMIENTOS *****
160 T1=X-Y/50 T2=TIME AB=ATN(X/2)
162 T1=ATN(.64FE) T2=SIN(R0-T1) T3=(.64FE)*2+140/140 T3=SORT(T3)*T2/R3
164 IF ABS(T3) THEN 170
166 WH=FE*(P1-2*AB) IF SGN(T3)=1 THEN AF=P1/2 ELSE AF=P1/2
168 GOTO 172
170 AF=ATN(T3/SORT(T3+1)) WH=ABS(EF*(R0-AF)) AF=SGN(AF)*(PI-ABS(AF))
172 AA=2*AF-A0 IF LY=136 THEN AA=3*PI-AA ELSE AA=PI-AA
174 CA=COS(A0) SA=SIN(A0)
176 SA=SIN(AA) CA=COS(AA)
178 V1=CA*RC+CA*MM V2=-SA*MSA+MM*HEF
180 V3=CA*CA*MM V4=V2-SA*SA*MM*HEF
182 U1=ABS(V1)+U2=ABS(V2) U3=ABS(V3) U4=ABS(V4)
184 IF U1/2 THEN 192
186 IF U1/2 THEN 198
188 IF U1 THEN I=4 GOTO 202 ELSE J=1
190 V=U1 I=SGN(V1) C3=V2 U1 C1=V3 U1 C2=V4 U1 GOTO 204
192 IF U2 THEN I=56
194 IF U2/4 THEN J=4 GOTO 202 ELSE J=2
196 V=U2 K=SGN(V2) C3=V1 U2 C1=V3 U2 C2=V4 U2 GOTO 204
198 IF U3 THEN I=4 GOTO 202 ELSE J=3
200 V=U3 I=SGN(V3) C3=V4 U3 C1=V1 U3 C2=V2 U3 GOTO 204
202 V=U4 I=SGN(V4) C3=V3 U4 C1=V1 U4 C2=V2 U4 GOTO 204
204 I=INT(RB/5) V=V/20+5
206 REM 6 MOVIMIENTO ANTES DEL CHOQUE *****
208 T=0
210 X=X+V1 Y=Y+V2 INT(T*(X-Y)+5)
212 X=X+E1 Y=Y+E1 X4=X+E1 Y4=Y+E1
214 IF X3=0 OR X4=X4 THEN X=X0 Y=Y0 GOTO 132
216 FOR IL=0 TO IL NEXT
218 D=X2-X*(X2-X)+(Y2-Y*(Y2-Y) IF D>R4 THEN PUT(X3,Y3)-(X4,Y4),P,PSET GOTO 2
10
220 T5=TIMER
222 PUT(XB,YB)-(X+RB,YB+E),B,OR
224 T1=X-Y/50
226 REM 7 MOVIMIENTO DESPUES DEL CHOQUE *****
228 GOSUB 84

```

posteriormente. Esta opción puede suprimirse fácilmente al adaptar el programa a otro ordenador de menor memoria.

El ordenador mediante su reloj interno puede medir el tiempo durante el que se mueven los discos, antes y después del choque, y de esta forma calcular sus velocidades. La comparación de estos valores con los datos teóricos que se calcularon previamente al choque, se realiza mediante la tabla de autoevaluación (opción 9) y permiten una primera estimación de la bondad de la simulación.

Obtención de IR

Si se modifica el programa será probablemente necesario variar la

sea VV respecto de CO, mayor habrá de ser el valor de IR. En el caso de choque frontal VV/CO sólo depende de la masa del blanco.

La expresión que nos da los valores de IR en función de los de VV/CO se ha conseguido ajustando los pares de valores obtenidos mediante una recta, que llamaremos, recta de calibrado (línea 204).

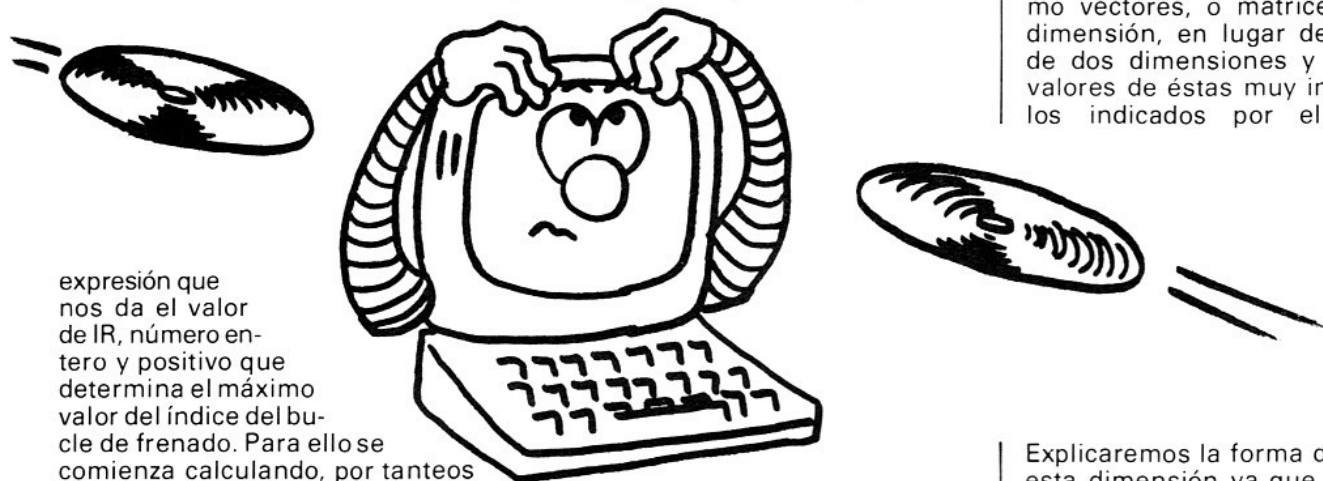
Como el tiempo que el ordenador emplea en ejecutar la sentencia PUT depende de las dimensiones de la matriz, es decir, del tamaño del disco, hay que obtener una recta de calibrado para cada uno de los 13 posibles tamaños del disco que corresponden a los respectivos valores de RB (línea 120). Como todas estas rectas de calibrado tenían una ordenada en el origen muy similar, hemos

dan los valores de IR y de VV/CO y se pide un nuevo valor de IR. Si una vez elegida esta opción se desea volver al menú es suficiente dar a IR el valor 0.1. El valor de IR también puede introducirse al comienzo del programa o al dar unos nuevos valores de los parámetros (opción 2).

En el caso de que sea necesario ajustar JR se hará de forma similar. En ninguno de los dos ajustes es necesario considerar valores de M mayores de tres, ya que en este caso VV=0.5, el menor valor posible.

Nota sobre las dimensiones de P y B

Las matrices P y B que permiten guardar los discos proyectil y blanco se han dimensionado como vectores, o matrices de una dimensión, en lugar de matrices de dos dimensiones y con unos valores de éstas muy inferiores a los indicados por el manual.



expresión que nos da el valor de IR, número entero y positivo que determina el máximo valor del índice del bucle de frenado. Para ello se comienza calculando, por tanteos sucesivos, el valor de IR, en el caso de choque frontal; para un valor dado del radio del blanco y para diferentes valores de su masa (por ejemplo, $M=0.01, 0.1, 0.25, 0.5, 0.75, 1, 2$ y 3). Gracias al diseño del programa, una vez ajustada la velocidad en el caso choque frontal, queda automáticamente ajustada en los demás casos posibles.

El movimiento del disco proyectil antes del choque se consigue aumentando sucesivamente en una unidad la componente horizontal de su centro. El valor de la componente vertical se calcula de modo que el desplazamiento del disco sobre la pantalla se ajuste lo mejor posible a la recta seleccionada por el joystick. Por lo tanto, la rapidez del proyectil antes del choque está relacionada con la componente horizontal de su velocidad definida por la variable CO (línea 174).

Los movimientos del proyectil y del blanco después del choque, vienen regidos de forma similar por el valor de la mayor de las componentes de las dos velocidades, definida por la variable VV (líneas 190 a 202). Cuanto mayor

promediado estos valores obteniendo el de -83. Ahora se ajustan unas nuevas rectas de calibrado cuya ordenada en el origen sea -83. Las pendientes de estas rectas vienen dadas en la línea 98.

Si sólo se desea estudiar cualitativamente el choque, no es necesario realizar este calibrado, pudiéndose considerar $IR=0$. También puede calibrarse sólo para algunos valores de R.

Para obtener los valores de IR que permiten calcular las rectas de calibrado, es conveniente realizar las siguientes modificaciones en el programa presentado:

- Inutilizar o eliminar las líneas 98 y 100.
- Sustituir 364 por 132 en la línea 482, INPUT M por IMPUT M, IR en la 394 y toda la línea 204 por REM.
- Introducir las líneas 119 INPUT «IR=»; IR y 481 PRINT «IR=»; IR, «VV/CO=»; VV/CO: INPUT «IR=»; IR: IF IR=.1 THEN 366 132.

De esta forma, al final de la tabla de autovalores (opción 9) se

Explicaremos la forma de obtener esta dimensión ya que su justificación es que el programa funciona correctamente y con un gran ahorro de memoria.

Cada dimensión de una matriz supone 5 bytes de memoria, es decir, 40 bits, mientras que un solo bit nos permite determinar el estado de cada punto de la pantalla en PMODE4. Por lo tanto, cada dimensión nos permitirá guardar el estado de 40 de estos puntos. Teniendo en cuenta que la sentencia GET es

GET(X1,Y1)-(X2,Y2),B

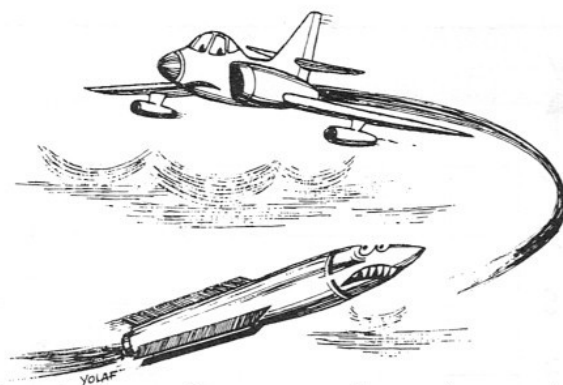
las dimensiones del rectángulo definido es $(X2-X1+1) \cdot (Y2-Y1+1)$, ya que hay que considerar los dos puntos extremos de cada segmento. Para guardar este rectángulo definiremos B como una matriz cuya única dimensión viene dada por:

$INT((\text{área del rectángulo}-1)/40)$
 $=INT((X2-X1+1) \cdot (Y2-Y1+1)/40)$
 Recuérdese que la enumeración de los índices comienza por cero.

Christian Wagner (*)

(*) Doctor en Ciencias Físicas. Catedrático de Física y Química del I. B. Mixto de Arcos de la Frontera (CADIZ).

Avión espía para-TI 57



La TI 57 es lo suficientemente potente como para mandarla en misión aérea. ¡Cuidado con los misiles!

Desde hace poco, la potencia bélica enemiga despliega sobre el teatro de operaciones unas instalaciones de defensa aérea terriblemente eficaces, que impiden la penetración de los bombardeos en un perímetro muy importante (Fig. 1).

En cuanto penetra un avión en esta zona, lo percibe el dispositivo radar de la base anti-aérea y un misil de cabeza buscadora encerrado en un silo (coordenadas $x=5$, $y=5$) es lanzado automáticamente e intenta destruirlo. Su misión consiste en sobrevolar uno de estos nuevos dispositivos anti-aéreos para fotografiarlo a baja altitud y volver ileso a su base con las fotos.

Mientras dure esta operación, el misil le perseguirá, pero su avión-espía está equipado con un radar de a bordo, que, con excelente precisión, le comunicará la posición del misil. Cuanto más se aproxime dicha posición a la de su avión, tanto más estará amenazado. Si las dos posiciones son estrictamente idénticas, significa que el misil le ha alcanzado (*auténtica Perogrullada*). Seguramente le condecorarán a título póstumo, y nadie verá nunca las fotografías del dispositivo enemigo.

El avión-espía se desplaza de una casilla a la contigua, según el eje "x" o el eje "y", o bien en diagonal (Fig. 2). Por lo tanto, en todo momento, puede escoger entre ocho direcciones posibles.

La pista desde donde despegue está situada en el borde del perímetro prohibido; por lo menos una de sus coordenadas es obligatoriamente ϕ o 1ϕ . Es en esta misma pista donde debe volver a tomar tierra, después de haber sobrevolado en diagonal

una de las cuatro casillas que rodean el silo del misil.

Listado del programa

Nº de paso	Código	Tecla
00	32 1	STO 1
01	81	R/S
02	32 2	STO 2
03	33 1	RCL 1
04	65	—
05	33 3	RCL 3
06	85	=
07	22	$x \leq t$
08	33 2	RCL 2
09	65	—
10	33 4	RCL 4
11	85	=
12	— 27	$R \rightarrow P$
13	32 6	STO 6
14	22	$x \leq t$
15	32 5	STO 5
16	01	1
17	83	.
18	02	2
19	32 7	STO 7
20	33 5	RCL 5
21	— 76	$x < t$
22	48 0	Fix 0
23	19	C.t
24	33 0	RCL 0
25	22	$x \leq t$
26	33 6	RCL 06
27	27	$P \rightarrow R$
28	34 4	SUM 4
29	33 4	RCL 4
30	81	R/S
31	22	$x \leq t$
32	34 3	SUM 3
33	33 3	RCL 3
34	81	R/S
35	48 1	FIX 1
36	71	RST
37	86 1	Lbl 1
38	— 19	INV C.t
39	48 1	Fix 1
40	05	5
41	32 4	STO 4
42	32 3	STO 3
43	15	CLR
44	81	R/S
45	32 0	STO 0
46	15	CLR
47	81	R/S
48	71	RST

Queda sólo paso libre para posibles mejoras. . .

En cuanto haya despegado, está localizado y el misil le persigue. Cualesquiera que sean sus ganas de huir, no olvide que tiene que sobrevolar una de las casillas de la base enemiga (para fotografiarla), antes de poder refugiarse en su punto de salida.

En la TI 57, el programa ocupa 49 pasos (0 a 48). En el paso 45, se coloca en la memoria 0, el nivel de dificultad (1,10 a 1,50). En los pasos 00 y 03, las coordenadas (x,y) de su avión están almacenadas, respectivamente, en las memorias 1 y 2. Al principio del juego estas coordenadas son las de su pista de despegue: Es ahí donde tendrá que volver para acabar su misión, es vital no olvidarlas.

El resto del programa calcula y visualiza las coordenadas del misil, después de cada uno de sus desplazamientos.

Para empezar el juego: **SBR 1**. La TI 57 visualiza 0. Debe introducir el nivel de dificultad deseado (de 1,10 a 1,50) y pulsar R/S. De nuevo 0 en pantalla. Debe indicar la abscisa de su pista (presione R/S) y luego la ordenada.

Un tercer toque de la tecla R/S inicia ahora el cálculo y visualización de la **ordenada** del misil, con una precisión de un decimal. Nueva pulsación de R/S y mismo tratamiento para la **abscisa** del misil, aquí también la precisión es de un decimal.

Introduzca la nueva posición de su avión, recordando que los únicos desplazamientos autorizados son los de la figura 2. La calculadora le da entonces la nueva posición del misil (ordenada y luego abscisa), y así sucesivamente.

Importante: La eficacia del misil es tan temible que no se puede navegar con aproximaciones. Hábrá que disponer de una tabla

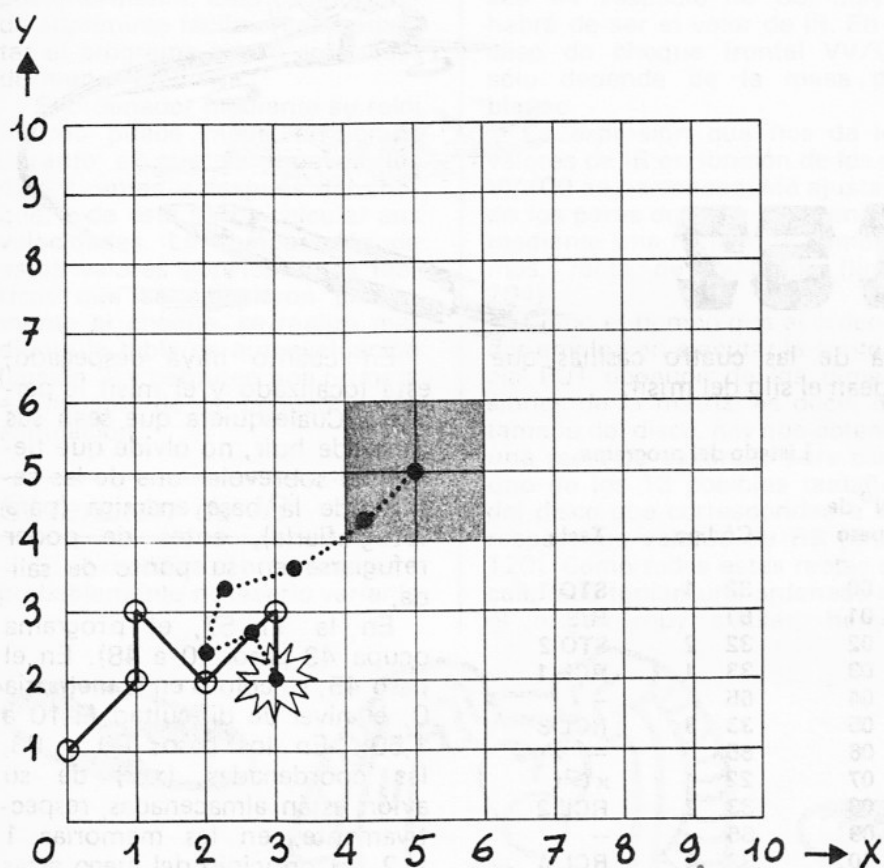


Fig. 1: En trazo continuo la trayectoria del avión, en punteado la del misil. En 5,5 el silo del misil.

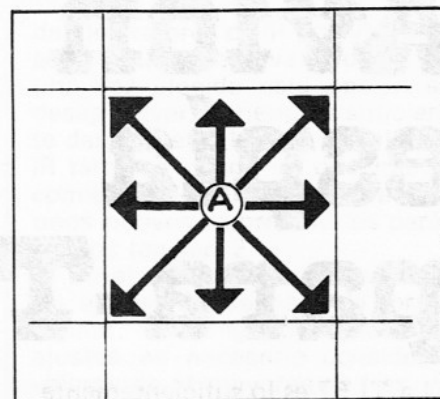


Fig. 2: Cuando el avión-espía se encuentra en A, hay que escoger entre ocho direcciones.

10x10 que represente las cien casillas de la zona prohibida, en la que se tracen con dos colores distintos, las trayectorias del avión y del misil.

Por último, no olvide que el misil puede alcanzarle, in extremis, en la pista de aterrizaje. Una vez que haya introducido las coordenadas de la base (Uf!), tenga el valor de mirar las del misil... Tranquilícese: Ya se han visto supervivientes....

Yvan Yourassowsky

7 líderes en su sector

Departamento de Publicidad:
 Ferraz, 11. Madrid - 8.
 Téls.: (91) 241 34 00 - 247 30 00.

Supervisión de cuentas corrientes con un Sharp PC-1211

Este pequeño programa de gestión, será muy útil para ordenar cantidades a partir de un valor numérico que puede ser una fecha, dando después un balance en cada una de las cantidades.

El programa, prevee que vamos a supervisar los movimientos de tres bancos y así lo primero que hace cuando le damos en el modo DEF el comando **SHFT C**, es preguntarnos por el saldo inicial que tenemos en cada uno de los tres; o sea SAL 1, SAL 2, SAL 3, que daremos respectivamente con las cantidades adecuadas. También nos pregunta el AÑO en que deseamos situar estos movimientos, cosa que hace con la pregunta **A # 0** con lo que le responderemos sólo con 2 cifras, por ejemplo 83 para el presente año 1983.

Después de esto, nos pide BANCO y FECHA en la forma FECHA, que hemos de responder con 5 cifras B XY MN donde B es el banco en que se produce el movimiento que puede ser 1, 2 ó 3, XY es el número del mes que será 01 para Enero, 02, para febrero hasta 12 para diciembre, por último MN es el día en que se produce el movimiento que será desde 1 a 31.

A continuación nos pedirá la CANTIDAD, en la forma **CANT**,

pero hay que darla con signo cambiado, es decir, ingresos como negativas (por comodidad, ya que suele haber más salidas que entradas).

Si deseamos ver cuál fue la última entrada, en lugar de **FECHA** le damos Exp9 (ENTER).

También en caso de confusión, cuando queramos repetir una entrada, se hace con sólo (ENTER) con lo que borra la última entrada que se hizo.

Una vez terminada la entrada de movimientos, por medio de **SHFT B**, se ejecuta un programa que ordena los movimientos por Banco en primer lugar y por fecha en segundo lugar, dando el balance de cada movimiento. Para esto es conveniente utilizar la impresora.

Para introducir más movimientos, podemos hacerlo con **SHFT A**.

Si deseamos listar de nuevo la salida de movimientos, sin necesidad de ordenarlos, haremos **SHFT L**.

Para grabar los datos a cassette haremos **PRINT#«8301»**; H

en caso de tratarse por ejemplo del primer trimestre del año 83.

Para recoger desde el cassette **INPUT#«8301»**; H.

```

10:PRINT "ERROR
":STOP
100:"C" CLEAR :
    INPUT "SAL1
    ";M;"SAL2 ";
    N;"SAL3 ";0;
    "ANO ";H
105:"A":IF I>=1
    LET D=E-1:
    PAUSE "ANTER
    .";A(D)
110:E=17+I:F=67+
    I:INPUT "FEC
    HA ";A(E):I=
    I+1:GOTO 112
111:I=I-1:GOTO 1
    05
112:IF A(E)=E9
    LET I=I-1:
    GOTO 105
120:INPUT "CANT.
    ";A(F):A(F)=
    -A(F):GOTO 1
    26
126:IF I=50GOTO
    170
130:IF I>=45GOTO
    150
140:GOTO 110
150:PAUSE "I,50:
    GOTO 110
  
```

```

170:"B"FOR B=17
  TO I+15
180:FOR C=B+1TO
  I+16
190:IF A(B)<=A(C)
  )GOTO 210
200:D=A(B):A(B)=
  A(C):A(C)=D:
  E=B+50:F=C+5
  O:D=A(E):A(E)
  )=A(F):A(F)=
  D
210:NEXT C
220:NEXT B
310:"L"FOR B=17
  TO I+16
320:C=INT (A(B)/
  E4):A=B+50:
  IF C=0GOTO 1
  0
325:IF B=17GOTO
  355
330:IF C<>FGOTO
  350
335:P=P+A(A):D=A
  (B)-E4*C:
  PRINT " " ";D
  "; "I(A)":
  PRINT P:F=C:
  GOTO 360
340:P=P+A(A):
  PRINT E4*H+A
  (B)-E4*C:"
  "I(A)":PRINT
  P:F=C:GOTO 3
  60
350:D=9+F:A(D)=P
355:D=12+C:P=A(D)
  ):PRINT " ":
  PRINT "BANCO
  "":C:PRINT "
  "":F=C:GOTO
  340
360:NEXT B
370:D=9+F:A(D)=P
510:END
900:"M":FOR E=1
  TO 106
910:IF A(E)=0
  GOTO 930
920:PRINT "MEM "
  "E:" "I(A(E)
930:NEXT E
940:END

```

Es difícil que una persona tenga más de 50 movimientos en un trimestre, por lo que por ejemplo, sería una forma de organización, hacerlo por trimestres que se guardan en cassette con los nombres.

1er. trimestre del año 1983 8301
 2º trimestre del año 1983 8304
 3er. trimestre del año 1983 8307
 4º trimestre del año 1983 8310

Félix Gutiérrez

EJEMPLO ORDENADO

En esta parte describiremos cómo se desarrolla un ejemplo paso a paso.

Una vez el programa en la máquina, al teclear **SHFT-C** lo que nos pregunta la máquina es **SAL1** que significa Saldo 1, o sea, el saldo en el primer banco y le daremos 95406 a continuación después del enter por supuesto nos pregunta por el **SAL2** y le daremos por ejemplo 81283 y a continuación el **SAL3** al que contestaremos con 2265, después nos pregunta por el **a#0** y le daremos las dos últimas cifras por ejemplo de la primera entrada que le daremos en el formato **BXYMN** donde **B** es 1 2 ó 3 según el movimiento pertenezca al banco 1 2 ó 3, **XY** es el mes, o sea, 01 para enero, 02 para febrero y así hasta diciembre y por último **MN** que es el día en que se produjo el movimiento, empezaremos diciéndole **FECHA 30314**.

Acto seguido nos pide **CANT.** y le contestamos 280 teniendo en cuenta que esta cantidad fue extraída del banco, o sea, es negativa y así en este ejemplo hasta 27 movimientos.

Una vez introducidos todos, damos **SHFT-B** con lo que se ordenan por orden creciente de banco y fecha y termina listando movimientos con saldo en cada uno.

En el supuesto de querer obtener otra copia haremos **SHFT-L**, para introducir más movimientos lo podemos hacer con **SHFT-A** y después el mismo proceso anterior y por último para obtener un volcado de la memoria con **SHFT-M**.

Para grabar datos al cassette lo hacemos con **PRINT#<8101>; H** y para volverlos a copiar desde el cassette con **INPUT#<8101>; H**, teniendo en cuenta de que el número máximo de movimientos que se pueden almacenar es de 50, cada cual se lo organiza por trimestres, semestres u otra forma.

Si durante la entrada de datos olvidamos cual fue la última, con **Exp9** nos dice cuál fue la anterior para que podamos seguir y si hicimos una entrada errónea con **ENTER** nos borra el último movimiento.

Por último queda el listado obtenido con los datos introducidos en el ejemplo.

BANCO 1.

810114.	-30000.
	65406.
123.	-30000.
	35406.
129.	41738.
	77144.
209.	-40000.
	37144.
226.	42318.
	79462.
313.	-1671.
	77791.
317.	-1800.
	75991.
323.	-776.
	75215.
325.	-4240.
	70975.
327.	42318.
	113293.
327.	-1510.
	111783.
327.	-2000.
	109783.
328.	-45000.
	64783.

BANCO 2.

810121.	-42289.
	38994.
121.	66862.
	105856.
128.	-25000.
	80856.
130.	-30000.
	50856.
210.	-19000.
	31856.
224.	67804.
	99660.
225.	-60000.
	39660.
317.	-30000.
	9660.
327.	67804.
	77464.

BANCO 3.

810101.	-52.
	2213.
116.	-51.
	2162.
216.	-250.
	1912.
314.	476.
	2388.
314.	-280.
	2108.

Máquinas tragaperras

OP-FRUIT

para Spectrum 48K

Quizá más de una vez se haya quedado con las ganas de seguir jugando a la máquina tragaperras de su bar habitual, y no haya podido hacerlo porque se quedó sin dinero. Ahora tiene la posibilidad de convertir su SPECTRUM en la más moderna máquina tragaperras y jugar cuanto quiera sin preocuparse de su bolsillo.

El juego

Acomódese; tras cargarse el programa con LOAD "" aparece la publicidad. A continuación la máquina le invita a jugar a la vez que le advierte que sólo da el 64,4% de sus ganancias, advertencia que no le dejará mientras esté jugando; pero no se preocupe, es bastante más de lo que suelen ofrecer este tipo de máquinas. Además jugando los avances inteligentemente, Vd. podrá variar ese porcentaje a su favor.

Ante Vd. aparecerá una máquina tragaperras de las de tres ventanillas. En la parte inferior de la pantalla se le ofrece la siguiente información adicional:

Credit: es el número de partidas de las que dispone.

M. Gastadas: Las monedas totales que lleva gastadas desde que empezó a jugar a la máquina.

M. Ganadas: las monedas totales que lleva ganadas desde que empezó a jugar a la máquina.

Dif.: es el número neto de monedas ganadas o perdidas desde que empezó a

jugar. No es más que la diferencia entre M. Ganadas y M. Gastadas. Si es un número positivo irá ganando dinero, si no irá perdiendo.

Bajo estos rótulos aparecen estos valores en monedas y una línea más abajo los mismos valores en ptas., teniendo en cuenta que cada moneda tiene un valor de 25 ptas. Pues bien, ahora tiene las siguientes posibilidades:

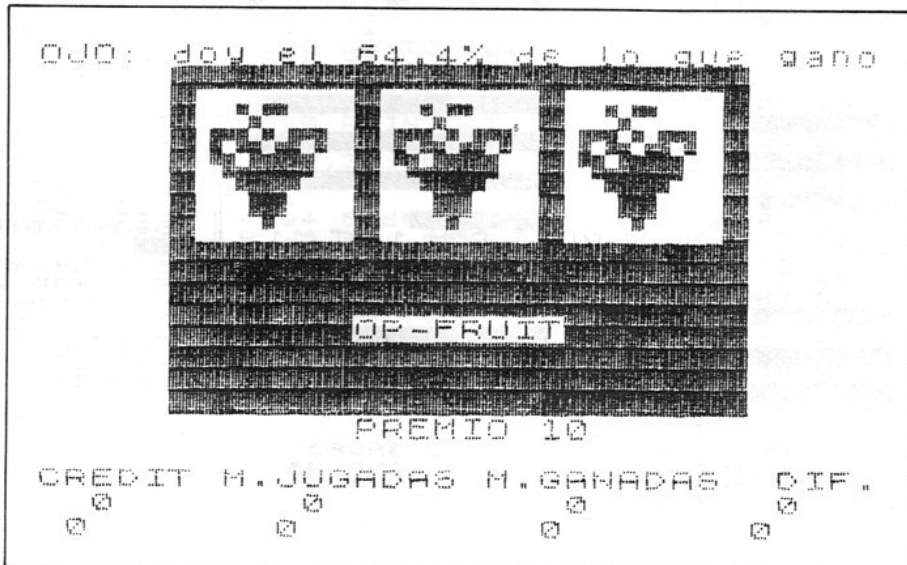
— Introducir una moneda de 25 ptas. en la máquina: esta

acción se simula pulsando la tecla P. Si la pulsa varias veces la máquina almacenará los créditos y podrá seguir jugando mientras no se le terminen.

— Tirar de la palanca: se simula pulsando la tecla B de su ordenador. En la máquina empezarán a oscilar las figuras y se fijará una combinación. Sólo funcionará si tiene créditos, si no introduzca una moneda en la máquina (es decir, pulse P).

— Poner los contadores a cero: pone a cero los contadores de M. Gastadas, M. Ganadas y Dif. (pero no el de Credit) para saber cuanto ganamos o perdemos desde este mismo momento. Se consigue pulsando la tecla O (letra, no el número cero).

— Destruir el programa: Pulsando T y siguiendo las instrucciones del mismo.



```

1 REM © 1984
2 BY Alberto Pajares Calleja
3 2 RANDOMIZE: BORDER 0: PAPER
6. CLS: PRINT INK 7; AT 12,1; ©
1984 BY EL Ordenador Personal
4 BEEP .3,7: BEEP .3,5: BEEP
.3,5: BEEP .7,4
5 BEEP .3,7: BEEP .7,14: BEEP
.3,12: BEEP .5,12: BEEP .7,11
PAUSE 10
6 BEEP .3,7: BEEP .3,5: BEEP
.3,5: BEEP .7,4
10 BEEP .3,7: BEEP .7,14: BEEP
.3,12: BEEP .5,11
11 BRIGHT 0: BORDER 7: PAPER 7
INK 2: CLS
12 PRINT AT 3,10: "Play"
13 INK 0
14 PRINT AT 7,2: " "
15 PRINT AT 8,2: " "

```

```

16 PRINT AT 9,2: " "
17 PRINT AT 10,2: " "
18 PRINT AT 11,2: " "
19 PRINT AT 12,2: " "
20 PRINT AT 13,2: " "
21 PRINT AT 14,2: " "
22 INK 1
23 PRINT AT 15,2: "Doy el 64.4%"
24 BEEP .4,20: BEEP .4,20: BEEP .4,20: BEEP .4,20

```

```

25 BEEP .4,20: BEEP .4,20: BEEP .4,20: BEEP .4,20
26 INK 0: BORDER 6: PAPER 6: C
27 PRINT INK 2: "Doy el 64.4% de lo que gano"
28 INK 1
29 PRINT " "
30 PRINT " "
31 PRINT " "
32 PRINT " "
33 PRINT " "
34 PRINT " "
35 PRINT " "
36 PRINT " "
37 PRINT " "
38 PRINT " "
39 PRINT " "
40 PRINT " "
41 PRINT " "
42 PRINT " "
43 PRINT " "
44 PRINT " "
45 PRINT " "
46 PRINT " "
47 PRINT " "
48 PRINT " "
49 PRINT " "
50 PRINT " "
51 PRINT " "
52 PRINT " "
53 PRINT " "
54 PRINT " "
55 PRINT " "
56 PRINT " "
57 PRINT " "
58 INK 3: PRINT FLASH 1; AT 11,12: "OP-FRUIT": INK 1
59 PRINT AT 11,20: " "
60 PRINT " "
61 PRINT " "
62 PRINT " "
63 PRINT " "
64 PRINT " "
65 PRINT " "
66 PRINT " "
67 PRINT " "
68 PRINT " "
69 PRINT " "
70 PRINT " "
71 PRINT " "
72 PRINT " "
73 INK 0
74 PRINT AT 17,0: "CREDIT"
75 PRINT AT 17,7: "M. JUGADAS"
76 PRINT AT 17,17: "M. GANADAS"
77 PRINT AT 15,12: "PREMIO"
78 PRINT AT 17,28: "DIF."

```

```

20 LET credit=1: LET premio=0
25 LET contador=-1: LET won=0
30 LET t=0: LET hp=0: LET tj=0
100 LET contador=contador+1
101 LET credit=credit-1
140 IF INKEY$="0" THEN LET contador=0: LET won=0
150 IF INKEY$="p" THEN LET credit=credit+1: GO TO 200
160 IF INKEY$="b" THEN GO TO 30
170 IF INKEY$="t" THEN GO TO 90
200 PRINT AT 16,2: " "
210 PRINT AT 15,2: credit
220 PRINT AT 15,10: contador
230 PRINT AT 15,20: won
240 LET dif=won-contador

```

```

250 IF dif<3000 OR dif>3999 THEN
  CLS: PRINT "Odeas empezar una
  nueva partida": PAUSE 200: GO TO
  0.45
255 PRINT AT 15,20: dif
257 PRINT AT 16,0: " "
260 PRINT AT 10,1: credit*25
265 PRINT AT 10,0: contador*25
270 PRINT AT 15,15: won*25
275 PRINT AT 15,27: dif*25
280 IF INKEY$=" " THEN GO TO 277
285 GO TO 130
300 IF credit>0 THEN GO TO 500
310 GO TO 130
320 IF hp>0 THEN LET num=200:
  GO TO 510
335 LET num=INT (RAND*1000)
340 LET n1=INT (num/100)
350 IF n1=0 OR n1=1 THEN LET a=
  1

```

```

530 IF n1=2 OR n1=3 THEN LET a=
  1.5
540 IF n1=4 OR n1=5 THEN LET a=
  2
550 IF n1=6 OR n1=7 THEN LET a=
  2.5
560 IF n1=8 THEN LET a=3
570 IF n1=9 THEN LET a=3.5
580 LET x=0: LET y=0
590 FOR n=0 TO 5
595 PRINT FLASH 1; AT x+n,y: " "

```

```

601 NEXT n
605 FOR w=0 TO 150: NEXT w
610 LET l=0
620 GO SUB a*1000
630 IF tj=1 THEN GO TO 4000
640 LET n2=INT ((num-n1*100)/10)
650 IF n2=0 OR n2=1 THEN LET b=
  1

```

```

660 IF n2=2 OR n2=3 THEN LET b=
  1.5
670 IF n2=4 OR n2=5 THEN LET b=
  2
680 IF n2=6 OR n2=7 THEN LET b=
  2.5
690 IF n2=8 THEN LET b=3
700 IF n2=9 THEN LET b=3.5
710 LET x=0: LET y=13
720 FOR n=0 TO 5
725 PRINT FLASH 1; AT x+n,y: " "

```

```

731 NEXT n
735 FOR w=0 TO 150: NEXT w
740 LET l=0
750 GO SUB b*1000
760 IF tj=1 THEN GO TO 4000
770 LET n3=num-n1*100-n2*10
780 IF n3=0 OR n3=1 THEN LET c=
  1
790 IF n3=2 OR n3=3 THEN LET c=
  1.5

```

```

740 IF n3=4 OR n3=5 THEN LET c=

```


Supongamos que ya tiene una combinación de figuras en la pantalla. Si le corresponde un premio la máquina reproducirá una melodía (tiene 13 diferentes, una para cada premio) y le indicará en monedas el valor del premio obtenido, valor que irá a engrosar su contador de monedas.

Si no es así la máquina, en un alarde de generosidad le ofrecerá una de cada 5 veces y aleatoriamente la posibilidad de avanzar las figuras de cualesquiera de las tres ventanillas. En este caso aparecerá en pantalla el rótulo AVANCE intermitentemente y acompañado de un sonido ululante. Oprima las teclas 1, 2 ó 3 según desee avanzar las figuras de las ventanillas izquierda, central o derecha. Si consigue premio enhorabuena por su astucia. Si no la máquina lo lamenta y puede seguir jugando.

Las figuras que pueden salir son: cerezas, limones, peras, fresas, campanas y la marca de la máquina, que de ahora en adelante notaré como M.M.

Las combinaciones premiadas son:

— — C:	1 moneda	C — cereza
— C C:	2 monedas	L — limón
C C C:	3 monedas	P — pera
L L L:	4 monedas	F — fresa
P P P:	6 monedas	Ca — campana
F F F:	10 monedas	
Ca Ca Ca:	20 monedas	
MM C C:	8 monedas	
MM L L:	12 monedas	
MM P P:	14 monedas	
MM F F:	16 monedas	
MM Ca Ca:	20 monedas	
MM MM MM:	20 monedas y repite premio hasta ganar 100 monedas en total	

El Programa

Comienza con publicidad y música de fondo (líneas 1 — 11). De la 12 a la 21 la máquina le invita a jugar y le advierte de lo generosa que es también musicalmente. Entre la 43 y la 79 está la presentación en pantalla de la máquina tragaperras y de la información adicional, y a continuación (80 — 96) está la inicialización de las variables. Las líneas 100 y 101 actualizan el número de monedas gastadas y el número de créditos disponibles. A continuación el programa analiza el teclado según las opciones que puede ofrecer al usuario y actualiza los distintos contadores (líneas entre



140 y 310). Luego el ordenador genera un número, aleatorio, de tres cifras mediante la función RND, asocia a cada valor de centenas, decenas y unidades la representación gráfica correspondiente y simula la rotación de las figuras en la pantalla (500 — 800).

Líneas 1000 — 1170: subprograma de la representación gráfica de las cerezas.

Líneas 1500 — 1620: Idem del limón.

Líneas 2000 — 2150: Idem de la pera.

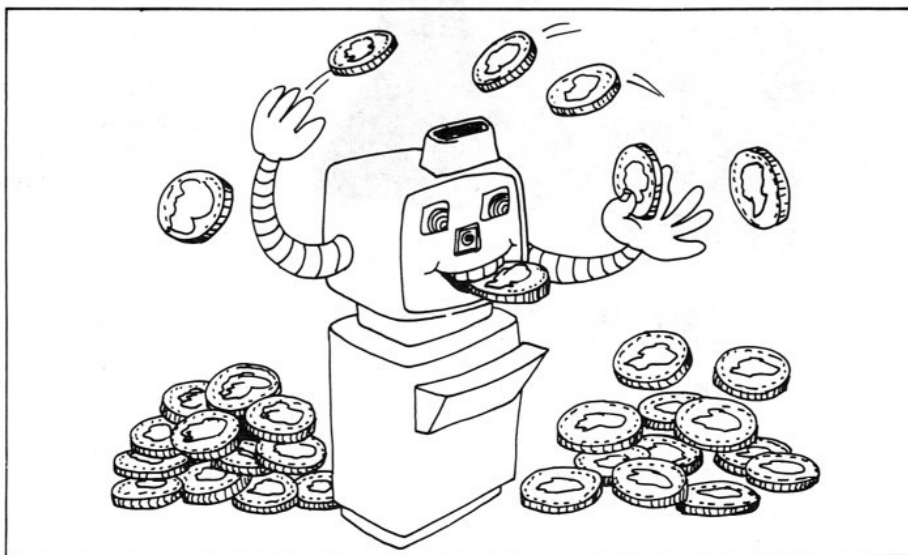
Líneas 2500 — 2680: Idem de la fresa.

Líneas 3000 — 3150: Idem de la campana.

Líneas 3500 — 3590: Idem de la marca de la máquina.

Entre las líneas 4000 y 5420 el SPECTRUM comprueba si la combinación resultante es premiada y si lo es ordena ejecutar una melodía distinta para cada premio. En este caso y tras la melodía se simula la caída de las monedas con un peculiar sonido (líneas 8000 — 8150). A continuación aparece el subprograma que permite destruir el programa principal con sólo apretar una tecla.

Las siguientes líneas hasta la 9140 analizan el teclado en el

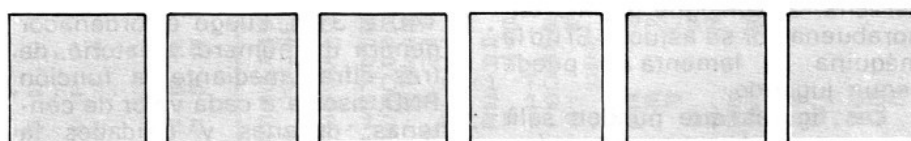
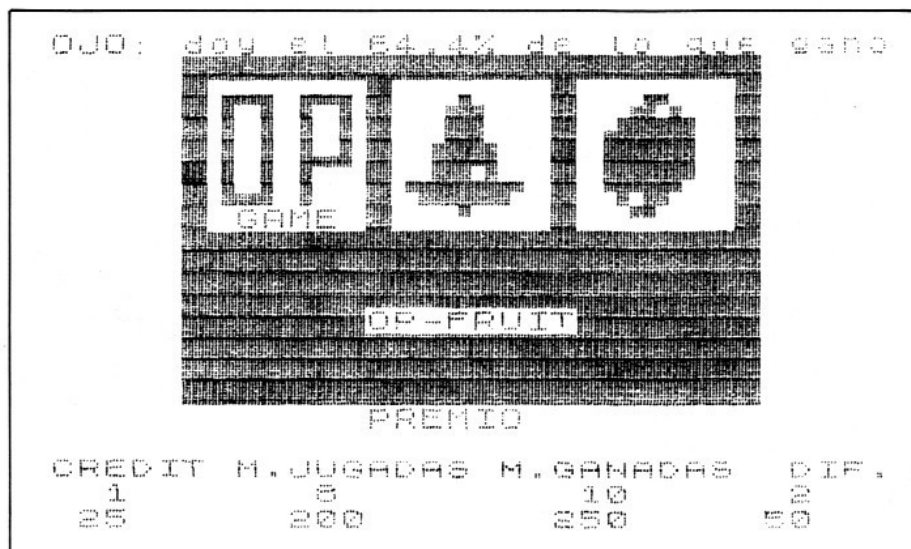


caso de que la máquina permita avanzar figuras en alguna de sus ventanillas, posibilidad que viene determinada por la línea 5400. Y por último la línea 9900 permite guardar en cinta al programa.

Anímesese y teclee el programa

Sabemos que su longitud no le asusta. A cambio su extrema simplicidad en el lenguaje le permitirá ir comprendiendo casi todos los pasos.

La máquina, aunque un poco avara, puede permitirle ganar algún dinero si juega con habilidad sus avances y sabe retirarse a tiempo. Si consigue fijar la marca de la máquina en las tres ventanillas habrá ganado 20 monedas (500 ptas.); atento y siga jugando pues esta combinación se le repetirá 4 veces más hasta que Vd. reune las 2.500 ptas. que constituyen el premio máximo. No obstante no confíe mucho en que le ocurra esto pues sólo tiene una posibilidad de cada 1000 (más las que le ofrezcan los avances); pero seguro que Vd. habrá descubierto que cortando el programa y tecleando:



LET núm; = 999: GO TO 510 conseguirá el codiciado premio sin necesidad de jugar ninguna partida. Cambiando el número 999 puede obtener cualquier combinación (¡pero está cometiendo un fraude!)

Gane o pierda dinero estamos seguros de que le gustará este

programa que aprovechará las posibilidades gráficas, color y sonido de su SPECTRUM y le creará adicción a las pocas partidas. Guarde el programa tecleando GO TO 9900.

Alberto Pajares Calleja

ORDENADOR PERSONAL Z-90



**EN KIT
O
MONTADO**

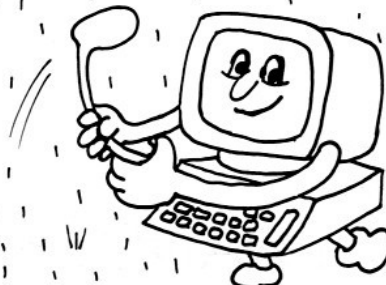
Unidad central incorpora 2 microprocesadores Z-80. 64K. RAM. Teclado alfanumérico más numérico, 8 teclas de funciones, pantalla de alta resolución. Tarjeta controladora de disco gobierna hasta 1,9Mb. ampliable hasta 10Mb. 3 puertas serie RS-232. Unidad de disco 160K. También incluye sistema operativo CP/M y lenguaje de programación Basic.

H-90-82 En Kit 7 Horas de fácil montaje que se ayuda con instrucciones	259.000 pts.
Z-90-82 Montado y probado	366.000 pts.
S-11 Impresora Honeywell 100 cps, 80 columnas	86.500 pts.

Comercial CRUZ

MONTESA, 38 - Telfs.: 402 92 41 - 401 26 26 - MADRID-6

MINIGOLF



Para los utilizadores de TRS 80 y Video Genie publicamos este programa de minigolf para que paseis unas veladas entretenidas.

```

4 CLEAR1000:CLS:RANDOM:FORX=0T0127:SET(X,0):SET(X,47):NEXT:FORY=0T047:SET(0,Y):S
ET(127,Y):NEXT:PRINT@67,STRING$(18,179):PRINT@131," M I N I G O L F":PRINT@195
,STRING$(18,179);
5 PRINT@899,"<@> 1980,COPYRIGHT BY KARNAK ELECTRONICS":FORT=1T02000:NEXT:CLS:HO
=1
6 IFHO>4THEN45ELSECLS:FORX=0T0127:SET(X,3):NEXT:FORY=3T047:SET(127,Y):SET(126,Y)
:NEXT:FORX=127T00STEP-1:SET(X,47):NEXT:FORY=47T03STEP-1:SET(0,Y):SET(1,Y):NEXT:L
$=STRING$(3,191)
7 PRINT@50,"HOYO NO.":HO;:ONHOGOSUB51,37,39,50
8 X=RND(124)+1:Y=RND(43)+3:IFNOTPOINT(X,Y)THENSET(X,Y):GOSUB33:GOSUB24ELSE8
9 A=1:B=1
10 P=X:Q=Y:SET(X,Y):IFK<PGTHENK=K+1:GOTO12
11 IFK>FGTHENGOSUB24ELSEK=K+1G:FORT=0TOK-PG:NEXTT
12 IFPOINT(X+A,Y+B)THENK=K+2*1G:GOSUB34:GOTO13ELSEX=X+A:Y=Y+B:RESET(P,Q):GOTO10
13 A1=A:A2=B:RESET(X,Y):IFPOINT(X,Y+B)THENB=-B
14 IFPOINT(X+A,Y)THENA=-A
15 IFNOTPOINT(X+A,Y+B)THEN10
16 IFRND(3)<>1THEN18
17 A=-A1:B=-A2:IFNOTPOINT(X+A,Y+B)THEN10
18 A=A1:B=A2:F=RND(2):ONFGOTO19,20,21,22
19 IFNOTPOINT(X+A,Y)THENX=X+A:GOTO10
20 IFNOTPOINT(X,Y+B)THENY=Y+B:GOTO10
21 IFNOTPOINT(X-A,Y)THENX=X-A:PRINT@0,3:GOTO10
22 IFNOTPOINT(X,Y-B)THENY=Y-B:PRINT@0,4:GOTO10
23 RETURN
24 K=0:PRINT@0,"ENTRE POTENCIA DEL GOLPE <0 A 9>";
25 A$=INKEY$:IFA$=""THEN25ELSEPG=VAL(A$)
26 IFA$="0"THENPP=NG+HO*20:H(HO)=PP:TG=TG+PP:NG=0:PRINT@0,STRING$(32,32):PRINT@
0,"PENALIZACION":20*HO;:FORT=1T01500:NEXT:HO=HO+1:GOTO6
27 IFPG<0ORPG>9THEN25ELSEPRINT@0,STRING$(35,32);
28 NG=NG+1:1G=(1/(PG+1))*10:PG=PG*20:FG=PG+60:IFA=1THENIFX>XPTHENA=-A:GOTO30
29 IFA=-1THENIFX<XPTHENA=-A
30 IFB=1THENIFY>YPTHENB=-B:RETURN
31 IFB=-1THENIFY<YPTHENB=-B:RETURN
32 RETURN
33 XP=RND(124)+1:YP=RND(43)+3:IFNOTPOINT(XP,YP)ANDNOTPOINT(XP+1,YP)THENSET(XP,YP)
:SET(XP+1,YP):RETURNELSE33
34 FORH=XPTOX+1
35 IFX+A=HANDY+B=YPTHENPRINT@0,"HOYO CONSEGUIDO EN";NG;"GOLPES":H(HO)=NG:TG=TG+
NG:NG=0:RESET(X,Y):FORT=1T010:RESET(XP,YP):RESET(XP+1,YP):FORN=1T050:NEXT:SET(XP
,YP):SET(XP+1,YP):FORN=1T090:NEXTN:NEXTT:HO=HO+1:CLS:GOTO6
36 NEXTH:RETURN
37 M$=STRING$(63,191):M1$=STRING$(4,32):M2$=STRING$(12,32):FORX=64T0384STEP64:PR
INT@X,M$:PRINT@X+576,M$:NEXTX:FORX=132T0324STEP64:PRINT@X,M2$:PRINT@X+576,M2$
:NEXT:PRINT@392,M1$:PRINT@648,M1$;
38 FORX=176T0368STEP64:PRINT@X,M2$:PRINT@X+576,M2$:NEXT:PRINT@436,M1$:PRINT@6
92,M1$:M=148:N=404:FORX=1T05:FORX=MTONSTEP64:PRINT@X,M1$:PRINT@X+512,M1$:NEX
TX:M=M+5:N=N+5:NEXTX1:RETURN
39 M$=STRING$(39,140):M1$=CHR$(191)+CHR$(188)+CHR$(176):M2$=STRING$(4,191)+CHR$(
188)+CHR$(176):M3$=STRING$(7,191)+CHR$(140):M4$=STRING$(4,191)+CHR$(143)+CHR$(13
1):M5$=CHR$(191)+CHR$(143)+CHR$(131)
40 M6$=CHR$(131)+CHR$(143)+M5$:M7$=CHR$(143)+STRING$(7,191)+CHR$(143):M8$=CHR$(1
76)+CHR$(188)+M1$:M9$=CHR$(188)+STRING$(7,191)+CHR$(188):N1$=CHR$(176)+CHR$(188)
+CHR$(191):N2$=N1$+STRING$(3,191)
41 N3$=CHR$(140)+STRING$(7,191):N5$=CHR$(131)+CHR$(143)+CHR$(191):N4$=N5$+STRIN
G$(3,191):PRINT@65,M4$:PRINT@129,M5$:PRINT@82,M7$:PRINT@148,M6$:PRINT@101,M7$
:PRINT@167,M6$:PRINT@121,N4$;
42 PRINT@188,N5$:PRINT@385,M1$:PRINT@416,M1$:PRINT@413,N1$:PRINT@444,N1$:PR
INT@449,M2$:PRINT@480,M2$:PRINT@474,N2$:PRINT@505,N2$:PRINT@513,M3$:PRINT@5
36,N3$:PRINT@544,M3$:PRINT@567,N3$;
43 PRINT@577,M4$:PRINT@602,N4$:PRINT@608,M4$:PRINT@633,N4$:PRINT@641,M5$:PR
INT@669,N5$:PRINT@672,M5$:PRINT@700,N5$:PRINT@897,M1$:PRINT@961,M2$:PRINT@9
16,M8$:PRINT@978,M9$:PRINT@935,M8$;
44 PRINT@997,M9$:PRINT@956,N1$:PRINT@1017,N2$:RETURN
45 CLS:PRINT"FIN DEL RECORRIDO":PRINTSTRING$(19,140):PRINT"EN TOTAL HA CONSEGUI
DO HACER";HO-1;"HOYOS EN";TG;"GOLPES":PRINT:PRINTSTRING$(63,129):FORK=1T0HO-1:PR
INT"HOYO NO.":K;
46 PRINTTAB(15)H(K);"GOLPES":NEXTK:PRINT:PRINTSTRING$(63,140):PRINT:PRINTTAB(30)
"PULSE <S> SI DESEA JUGAR DE NUEVO";
47 A$=INKEY$:IFA$=""THEN47
48 IFA$="S"THENRUNELSECLS:END
49 IFINKEY$=""THEN49ELSEEND
50 M$=STRING$(18,191):I=203:FORX=1T04:PRINT@I,M$:PRINT@I+24,M$:PRINT@I+448,M$:
PRINT@I+472,M$:I=I+64:NEXT:PRINT@521,STRING$(46,140):PRINT@513,STRING$(5,191)
:PRINT@570,STRING$(5,191):RETURN
51 X=196:FORJ=1T010:FORN=XTOX+640STEP64:PRINT@N,CHR$(191):NEXT:X=X+6:NEXT:RETUR
N

```



LIBROS EN FRANCES

Ferraz, 11 - 3º
Tel. 247 30 00 - Madrid-8

BOLETIN DE PEDIDO

1.700 Pts. Pratique du VIC
2.150 Pts. Outils Financiers
2.500 Pts. Pom's
1.700 Pts. Visical sur TRS-80
1.700 Pts. Exercices pour TRS-80
900 Pts. Boites a Outils pour PC-1500
1.700 Pts. Le Vic a l'Affiche

P.V.P.	Serie	Canti- dad.	Título.
Hors collection			
1.050 Pts.			Visa pour l'informatique.
1.400 Pts.			Mon Ordinateur.
1.450 Pts.			L'ordinateur individuel.
Collection "Matériels".			
1.450 Pts.	Azul		Comprendre les microprocesseurs.
1.450 Pts.	Verde		La découverte de l'Applesoft - tome 1.
1.450 Pts.	Verde		La découverte de l'Applesoft - tome 2.
1.450 Pts.	Azul		La pratique de l'Apple II - vol. I.
1.450 Pts.	Rojo		La pratique de l'Apple II - vol. II.
1.700 Pts.	Negro		La pratique de l'Apple II - vol. III.
1.700 Pts.	Verde		La découverte du Goupil.
1.450 Pts.	Azul		La pratique du TRS-80 - vol. I.
1.900 Pts.	Rojo		La pratique du TRS-80 - vol. II.
1.700 Pts.	Negro		La pratique du TRS-80 - vol. III.
1.450 Pts.	Azul		La pratique du MZ-80 K.
1.450 Pts.	Verde		La découverte du PET/CBM.
1.450 Pts.	Azul		La pratique du PET/CBM - vol. I.
1.700 Pts.	Rojo		La pratique du PET/CBM - vol. II.
1.700 Pts.	Verde		La découverte du VIC.
1.450 Pts.	Verde		La découverte de la TI-57.
1.700 Pts.	Verde		La découverte du PC-1211.
1.450 Pts.	Azul		La pratique du ZX81.
1.450 Pts.			Le petite livre du ZX81.
Collection "Langages"			
1.450 Pts.	Verde		Langages de programmation.
1.700 Pts.	Rojo		Comment programmer.
1.450 Pts.	Azul		Programmer en Fortran.
1.450 Pts.	Verde		Programmer en Basic.
1.450 Pts.	Verde		Programmer en L.S.E.
1.700 Pts.	Rojo		Programmer en Pascal.
1.700 Pts.	Negro		Le langage ADA.
Collection "Guides Pratiques".			
1.450 Pts.	Azul		L'APL sur TRS-80.
1.000 Pts.	Azul		La réalisation des programmes.
1.450 Pts.	Azul		LISP sur Apple II.
1.700 Pts.			Les graphiques sur TRS 80.

P.V.P.	Serie	Canti- dad.	Título
Collection "Mémentos".			
1.700 Pts.	Azul		Clefs pour le PET/CBM.
Collection "Programmes".			
1.700 Pts.	Azul		Etudes pour ZX81.
1.700 Pts.	Verde		Jeux, trucs et comptes pour PET/CBM.
1.700 Pts.	Verde		Récréations pour TI-57 - tome 1.
1.700 Pts.	Azul		Récréations pour TI-57 - tome 2.
1.700 Pts.	Azul		Variations pour PC-1211.
1.700 Pts.	Rojo		Modèles pratiques de décision - tome 1.
1.700 Pts.	Rojo		Modèles pratiques de décision - tome 2.
Edi Tests			
1.700 Pts.			Les systèmes à microprocesseurs.
1.700 Pts.			Mise en oeuvre du BUS IEEE 488.
Novedades:			
2.150 Pts.	Rojo		Programme HP-41.
1.900 Pts.	Verde		La decouverte du FX-702 P.
2.150 Pts.	Azul		Le BASIC de A à Z.
1.900 Pts.	Azul		Jeux, Trucs et comptes pour TRS-80.
1.700 Pts.	Verde		Visicalc sur Apple.
2.150 Pts.			La Comptabilité sur Apple II.
2.150 Pts.			Les finances familiales.
1.700 Pts.			Etudes pour ZX-81 (2)
1.550 Pts.			Pascal sur TRS-80
1.700 Pts.			La pratique du ZX-81 (2)
1.700 Pts.			La decouverte du TI-99/4A
1.900 Pts.			College Poquettes et maths.
2.500 Pts.			Logic Basic (pour une programma- tion structurée).
1.700 Pts.			Le systeme Unix (utilisation des commandes).
1.900 Pts.			Le systeme CP/M pour Z-80 (adapta- tion du BIOS et compléments).
1.700 Pts.			Le systema Pascal UCSD (1/organisa- tion générale).
2.250 Pts.			Le Basic et l'ecole T2.
1.900 Pts.			Decouverte du PC 1500.
1.700 Pts.			Un ordinateur personnel signe IBM.
1.700 Pts.			Clefs pour le VIC.
1.700 Pts.			CP/M mot par. mot.

Total Libros Importe Total Pts.

Modo de pago elegido: Cheque adjunto

☐ Transferencia Bancaria a nuestra cuenta n° 1912 del Banco de Bilbao, Ferraz, 42 - Madrid-8.
☐ Contra reembolso ☐ En este caso cobramos los gastos de correos originados en cada caso.

NOMBRE APELLIDOS
CALLE Nº TEL.
CIUDAD D.P. PROVINCIA

FECHA:

FIRMA:

Aprendizaje del BASIC en un instituto de Bachillerato

Cada vez son más frecuentes los comentarios y artículos que aparecen en las revistas especializadas defendiendo la introducción de los ordenadores en los centros de enseñanza. Pero también es verdad que la mayor parte de las veces se toca el tema de una forma ideal y generalizada (se debería... habría que...). En este artículo se pretende bajar a lo particular contando la modesta pero importante experiencia realizada en un centro de enseñanza de nuestro país.

Las dificultades iniciales

La experiencia se está llevando a cabo en el *Instituto de Bachillerato Mixto de Cangas (Pontevedra)*. Tengo noticias de que no somos el único Instituto que realiza estos trabajos y la seguridad de que dentro de muy pocos años todos los centros de enseñanza dispondrán del material suficiente para implantar la Informática como una asignatura más o como complemento y ayuda a otras disciplinas.

Cuando uno intenta llevar a la práctica una idea quizá pone demasiado subjetivismo en el asunto, y ve cosas maravillosas donde los demás ven una trivialidad. Así, el primer cometido fue tantear la opinión de la Dirección del centro y la de los propios compañeros. Y aquí he de decir que el proyecto fue magníficamente acogido, lo que quiero sinceramente agradecer desde estas líneas. Tanto interés se puso en el asunto que inmediatamente empezamos a barajar presupuestos para la adquisición del ordenador.

Ya sabemos el poco dinero del que disponen los centros de enseñanza, así que, ¿qué mejor que un ZX-81 por lo menos para empezar? El proyecto ya estaba en marcha.

Todo esto surgió con el curso ya empezado, los alumnos ya estaban distribuidos en grupos, los profesores ya teníamos nuestros horarios y no era cuestión ahora de sentar frente al teclado del Sinclair a los treinta y tantos alumnos que componen una clase. Habría que resolver el problema de otra forma. El aprendizaje del BASIC tendría que realizarse fuera de las horas de clase.

Faltaba entonces la prueba más crucial, proponer la iniciativa a los alumnos. A pesar de que el cursillo era totalmente voluntario, y fuera del horario habitual de clases, la propuesta fue unánimemente aplaudida; tanto que planteaba un problema para formar pequeños grupos en que sus horas libres coincidieran con las mías. Esa coincidencia se daba con los alumnos de COU.

De los tres cursos de COU de ciencias (unos 100 alumnos) se formaron grupos de ocho, de forma que cada grupo asistiera a cinco sesiones semanales de una hora de duración, atendiendo a tres grupos semanalmente.

Parecía inabordable el pretender enseñar todo el lenguaje BASIC en cinco horas, no había otra solución, pues, o bien se quedaban alumnos de la misma clase sin ver el ordenador, o bien yo hacía un montón de horas extras, y teniendo en cuenta que paralelamente, como comentaré al final, se llevaba a cabo otro cursillo para los profesores no había más remedio que buscar la solución en esas horas.

El cursillo

Por lo expuesto hasta ahora se deduce sin dificultad que no se podía ser muy ambicioso. No se trataba entonces de que cada alumno fuese un buen programador al finalizar el cursillo, sino solo de que cada uno tuviera una ligera, pero eficaz visión de lo que es un ordenador, de lo que es un lenguaje informático y sobre todo perderle el respeto a esa *caja mágica* y ver que las computadoras no son sólo para los grandes sabios, sino que se le puede coger el *gustillo* por lo menos en cinco horas.

Con éstos objetivos y limitaciones a la vista, se trataba de conseguir que desde el primer momento el alumno supiera lo que es un programa, y a través de él fuese conociendo las distintas palabras que componen las órdenes que se le dan a la máquina; incluso que él mismo fuese capaz de elaborar un programa similar al anterior para el próximo día, el cual a su vez pudiera

modificarse ligeramente con el fin de introducir algún nuevo concepto.

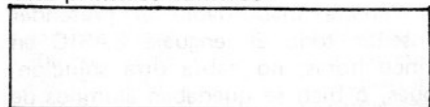
Considerando que los alumnos de COU están ya lo suficientemente *rodados* como para no poner mala cara a la introducción de unos conceptos nuevos, que aún con su lógica podían resultarles un poco chocantes, habría que proponer unos programas tales que centraran al alumno en los conceptos propios de la programación y no se perdieran en problemas ajenos a ellos, como podría ser un cálculo matemático excesivo, etc.

Programa núm. 1: TABLA DE MULTIPLICAR

Con este programa evidentemente el alumno no iba a perderse en mecanismos matemáticos propios de la multiplicación, sino que iba a prestar atención al cómo, a la forma de resolverlo, a las palabras clave de BASIC.

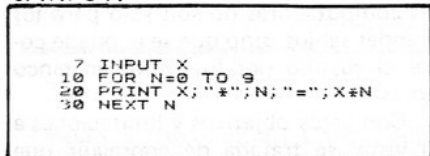
Después de una breve introducción al teclado y periféricos del ZX se intentaba comparar las posibilidades del micro-ordenador con la típica calculadora de bolsillo. Desde luego los dos realizan multiplicaciones, pero nuestro ZX sabe escribir palabras y puede presentar la tabla de multiplicar completa y con algún que otro adorno.

Inmediatamente confeccionábamos la tabla de multiplicar del 2 para los diez primeros números:



En este programa se introducían ya dos cosas importante, el PRINT y el ciclo FOR-NEXT. Se explicaba un poco como operaba la máquina con el ciclo y como distinguía dentro del PRINT lo que estuviese entre comillas (el ZX lo escribiría al pie de la letra) de lo que estuviese sin ellas (escribiría su valor numérico). Esto se entendía bien al comparar los dos $2 \cdot N$ que aparecen en el PRINT.

Una vez claros los nuevos conceptos pulsábamos RUN, y las caras se cubrían de una leve sonrisa de admiración y satisfacción. Pero surgía el *listillo*; "Si eso hay que hacerlo para cada número, ¡menudo rollo!". Vemos cómo ya el alumno buscaba la mejora. Habría que introducir un nuevo concepto: el INPUT.

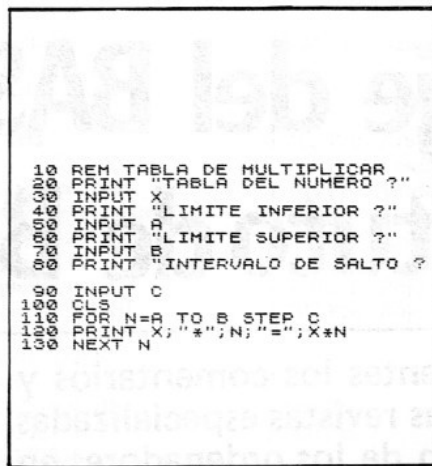


Se recalca con este nuevo programa la utilidad de espaciar las líneas de diez en diez, para poder tener sitio donde introducir nuevas cosas. Y sobre todo destacar como con la introducción de una sola línea (la 7) y la corrección de otra (la 20), se le sacaba mucho más rendimiento al programa,

pues ahora no sólo sabía la *tabla del 2*, sino la de cualquier número (X), incluso con decimales o negativo.

El siguiente paso era dar la regla de oro de la programación: "Todo programa ha de ser lo más versátil posible", o bien "a todo programa hemos de sacarle la mayor información posible". Entonces, ¿por qué fijar los límites del intervalo de la tabla (el 0 y el 9)? y ¿por qué limitar el salto de uno en uno?

Esta nueva mejora se resolvía introduciendo dos nuevos INPUTs y el STEP en el ciclo FOR-NEXT. El programa quedaría de forma definitiva:



Ahora aparecían dos nuevas sentencias, la REM de la línea 10 que no hace nada pero que es útil para poner anotaciones y títulos en el programa y el CLS de la línea 100 que significa *borrar la pantalla*, útil en este caso para que una vez introducidos los datos a través de los INPUTs, se borren esas preguntas y el resultado aparezca sólo en la pantalla (línea 120). Además; y esto es importante, ahora la máquina nos recuerda el tipo de dato que espera que le introduzcamos (líneas 20, 40, 60 y 80) para no perdernos en los INPUTs.

Era hora de hacer balance y recordar lo que habíamos aprendido:

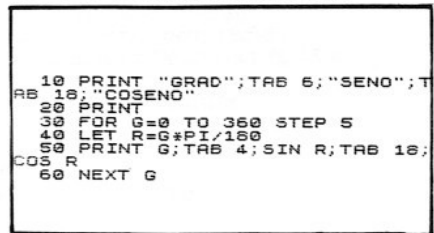
- 1) PRINT con sus:
 - Comillas (escribe al pie de la letra)
 - Sin comillas (escribe el valor)
 - Punto y coma (escribe al lado de lo anterior)
- 2) FOR-NEXT con su STEP (un ciclo)
- 3) INPUT, introducción de datos
- 4) REM, para recordatorios en el listado
- 5) CLS, para borrar la pantalla

Una vez llegados a este punto los alumnos tomarían las notas oportunas sobre lo explicado y se les proponería un ejercicio para el próximo día: Confección de una tabla trigonométrica para el seno y el coseno de forma similar a como habíamos hecho la tabla de multiplicar.

Esta tabla iría de cinco en cinco grados sexagesimales desde 0° a 360° . La dificultad residiría en que el ZX hace los cálculos en radianes.

Programa núm. 2: TABLA TRIGONOMETRICA

Los alumnos traerían su programa confeccionado individualmente o en grupos de dos. Una vez corregido y aclarado quedaría de la forma:



La novedad residiría en la línea 40, donde se guarda en la variable R la conversión de grados (G) a radianes para así poder simplificar la línea 50 donde hay dos llamadas a R, en el SIN y COS.

También hay que destacar los TABs de las líneas 10 y 50. El alumno entiende perfectamente que TAB N significa escribir en la columna N, lo mismo que la coma (,) lleva la escritura a la columna 16, en la mitad de la línea correspondiente. Estas instrucciones son útiles para ordenar y aprovechar el espacio de la pantalla. Finalmente decir que la línea 20 deja una línea en blanco para separar los títulos de los resultados.

Programa núm. 3: SISTEMA DE ECUACIONES

Previamente a la confección

Previamente a la confección del programa se habría resuelto a *mano* el sistema de ecuaciones general:

$$\begin{aligned} A1 X + A2 Y &= A3 \\ A4 X + A5 Y &= A6 \end{aligned}$$

cuyas soluciones serían:

$$\begin{aligned} X &= \frac{A6 - \frac{A5 A3}{A2}}{A4 - \frac{A5 A3}{A2}} \\ Y &= \frac{A3 - A1 X}{A2} \end{aligned}$$

Donde A1, A2, A3, A4, A5, A6 son los coeficientes que hay que introducir (mediante INPUTs) y X, Y las incógnitas a resolver.

Siguiendo lo aprendido en sesiones anteriores, los datos se introducirían de la forma:

```
10 INPUT A1
20 INPUT A2
30 INPUT A3
40 INPUT A4
50 INPUT A5
60 INPUT A6
```


Normalmente los alumnos ya no te dejan continuar. Inmediatamente surge la pregunta: "si tuviéramos un sistema de cuatro ecuaciones con cuatro incógnitas (20 coeficientes) ¿habría que poner 20 líneas para los INPUTs?" Evidentemente habría que mejorar la situación.

Introduciríamos aquí un nuevo concepto, el de variables con subíndices. Antes de introducirlas habría que dimensionar la variable con DIM (guardar un cierto número de subíndices para esa variable). Además, los subíndices irían entre paréntesis.

A pesar de la dificultad que entraña la introducción de este nuevo concepto, no hemos de olvidar que los alumnos de COU, en su asignatura de matemáticas, estudian los *Espacios Vectoriales* bastante a fondo y allí ya trabajan con los conceptos de *componentes* y *dimensión* de un vector. Sólo había que explicarles que A(1), A(2), A(3), A(4), A(5), A(6) son las componentes de un vector de dimensión seis DIM A(6). Los INPUTs los meteríamos dentro de un bucle que hiciera variar los subíndices:

```
10 PRINT "COEFICIENTES (POR FAVOR) ?"
20 DIM A(6)
30 FOR N=1 TO 6
40 INPUT A(N)
50 PRINT A(N); " ";
60 NEXT N
70 CLS
80 LET X=(A(6)-A(5)*A(3)/A(2))/A(4)-A(5)*A(1)/A(2)
90 LET Y=(A(3)-A(1)*X)/A(2)
100 PRINT A(1); "X+"; A(2); "Y="; A(3)
110 PRINT A(4); "X+"; A(5); "Y="; A(6)
120 PRINT
130 PRINT "SOLUCIONES: "
140 PRINT "X="; X
150 PRINT "Y="; Y
```

Además de repasar conceptos anteriores habría que destacar el punto y coma (;) en que termina la línea 50, que no tiene otro sentido más que escribir los A(N) unos al lado de los otros separados por un espacio en blanco.

Se fijaría también la atención en el juego de paréntesis de las líneas 80 y 90, que por otra parte actúan de forma idéntica a como lo hacen en las matemáticas.

Ahora se propondría como ejercicio para el día siguiente la realización de un programa que resolviera la ecuación de segundo grado.

Programa núm. 4: ECUACION DE 2º GRADO

Por supuesto que un alumno de COU ha de saber que las soluciones de una ecuación de la forma: $A X^2 + B X + C = 0$. Son:

$$X1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

$$X2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}$$

Traerían entonces un programa, como ejercicio, que en general se pareciera al siguiente:

```
10 PRINT "COEFICIENTES ?"
20 INPUT A
30 INPUT B
40 INPUT C
50 LET X1=(-B+SQR(B**2-4*A*C))/(2*A)
60 LET X2=(-B-SQR(B**2-4*A*C))/(2*A)
70 PRINT A;"X2=";B;"X=";C;"="0"
80 PRINT "SOLUCIONES:"
90 PRINT "X1=";X1;"X2=";X2
```

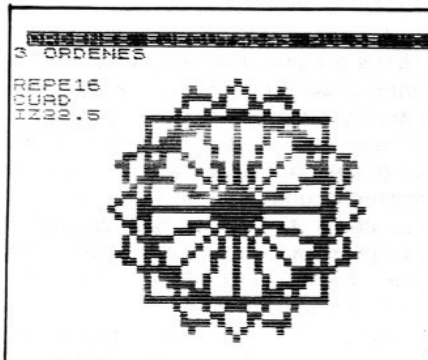
Lo probábamos con unos datos como éstos: A = 2, B = 3, C = 1 y efectivamente el programa funcionaba, los chicos quedaban muy satisfechos pues ya sabían programar.

En este punto era necesario aconsejarles que tuvieran prudencia. El que hubiera salido bien para unos datos muy concretos no era suficiente para generalizar. Efectivamente, se proponía ahora introducir los mismos datos anteriores pero la B negativa, B = -3. La decepción era evidente; el ZX daba un informe de error. ¿Dónde estaba el fallo?

Habría que aclarar que el ZX no era perfecto. No sabe elevar números negativos al cuadrado. Tendríamos que cambiar el término B**2 de las líneas 50 y 60 por B*B que hace lo mismo y no crea problemas, pues sí que sabe multiplicar números negativos.

Por inercia, esta dificultad felizmente resuelta, les traía a la memoria que una ecuación de segundo grado podía dar resultados imaginarios, es decir, que el término de la raíz cuadrada podía ser negativo. Si nosotros no sabemos hacer raíces de números negativos (supongo que se preguntarían), ¿sabría hacerlas el ZX?. Al probar con A = 4, B = -3, C = 1 efectivamente daba error, estaba claro que no sabía.

Ahora la modificación sería más profunda. Habría que analizar el discriminante (el radicando de la raíz) y enviar los cálculos a otro lugar del programa donde le arreglaríamos las cosas. Para ello sería conveniente también diferenciar la posible parte real de la imaginaria en las soluciones. Una vez expuestas las dificultades el programa quedaría definitivamente:



```
10 PRINT "COEFICIENTES ?"
20 INPUT A
30 INPUT B
40 INPUT C
50 CLS
60 PRINT A;"X2=";B;"X=";C;"="0"
70 PRINT
80 PRINT "SOLUCIONES:"
90 PRINT
100 LET D=B*B-4*A*C
110 LET R=-B/(2*A)
120 LET I=(SQR(ABS D))/(2*A)
130 IF D<0 THEN GOTO 170
140 PRINT "X1=";R+I
150 PRINT "X2=";R-I
160 STOP
170 PRINT "X1=";R;"I=";I
180 PRINT "X2=";R;"I=";-I
```

Fundamentalmente en este programa aparecen dos nuevos conceptos, el IF de la línea 130 y el STOP de la 160; este último no ofrece dificultad de comprensión pues significa que cuando el programa llegue ahí, se pare, es decir no se ejecutarían las líneas 170 y 180. El IF merece un poco más de atención.

Con el IF la máquina piensa, toma decisiones por sí misma, esto es fundamental en un ordenador. Fijémonos entonces en las dos opciones que presenta el discriminante (D).

Si D es positivo las cosas ocurren como en el programa que los alumnos habían confeccionado, pues la línea 130 no actuaría y el programa se detendría por STOP en la 160.

Ahora bien, si (IF) D es negativo (D<0) entonces (THEN) se va (GOTO) a la línea 170. No es que el ZX sepa resolver las raíces negativas, sino que le hemos arreglado las cosas en la línea 120 donde hace la raíz cuadrada del valor absoluto de D (ABS D hace que la máquina se olvide del signo de D, tomándolo en todo caso como positivo a pesar de ser D negativo). El arreglo consiste entonces en la forma de presentar la solución. Si D<0 el programa salta a las dos últimas líneas, donde escribe las soluciones separando la parte real de la imaginaria (la "I" es el símbolo matemático de $\sqrt{-1}$ y representa la parte imaginaria de un número).

Programa núm. 5: CONJUGACION DE VERBOS REGULARES

Con este último programa se pretendía demostrar que un ordenador no sólo sabe de matemáticas, sino que también puede ser útil en otras disciplinas. Al mismo tiempo introduciríamos el concepto de las cadenas, es decir, nuestras variables no serían números sino palabras.

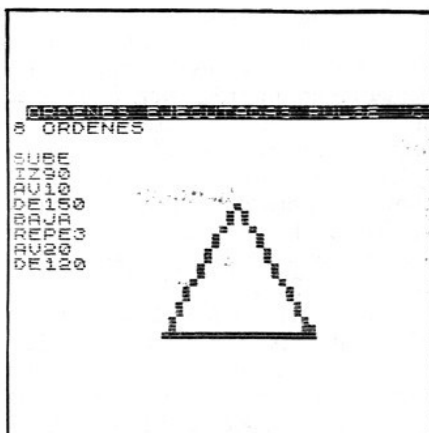
Naturalmente antes de escribir el listado hay que saber qué vamos a programar. Hemos de analizar y estructurar el problema a fondo.

Supongamos que queremos analizar un verbo de la primera conjugación, como el verbo AMAR. Escribamos entonces el Presente y Pretérito Imperfecto del modo indicativo:

Presente		Pretérito Imperfecto	
YO	AM O	YO	AM A BA
TU	AM AS	TU	AM A BAS
EL	AM A	EL	AM A BA
NOSOTROS	AM AMOS	NOSOTROS	AM A BAMOS
VOSOTROS	AM AIS	VOSOTROS	AM A BAS
ELLOS	AM AMAN	ELLOS	AM A BAN

El problema ya estaba claro. Todos los tiempos de los verbos empiezan por los pronombres (YO, TU, EL ...). Todos tienen la raíz, que resulta de suprimir al infinitivo (AMAR) las dos últimas letras, la terminación (AR). La raíz sería en nuestro caso AM.

El Pretérito Imperfecto tiene la AB para todas las personas. La terminación de los dos tiempos es idéntica, salvo para la primera persona del singular, pues mientras que el Presente es "O" en el Pretérito es "A".



```

10 DIM P$(6,9)
20 DIM T$(5,4)
30 LET P$(1)="YO"
40 LET P$(2)="TU"
50 LET P$(3)="EL"
60 LET P$(4)="NOSOTROS"
70 LET P$(5)="VOSOTROS"
80 LET P$(6)="ELLOS"
90 LET T$(1)="O"
100 LET T$(2)="AS"
110 LET T$(3)="A"
120 LET T$(4)="AMOS"
130 LET T$(5)="AIS"
140 LET T$(6)="AN"
150 PRINT "VERBO ?"
160 INPUT V$
170 LET L=LEN V$
180 LET R$=V$( TO L-2)
190 CLS
200 PRINT "PRESENTE"
210 PRINT
220 FOR N=1 TO 6
230 PRINT P$(N);R$;T$(N)
240 NEXT N
250 PRINT
260 PRINT "PRETERITO IMPERFECTO"
270 PRINT
280 FOR N=2 TO 6
290 PRINT P$(N);R$;"AB";T$(N)
310 NEXT N

```

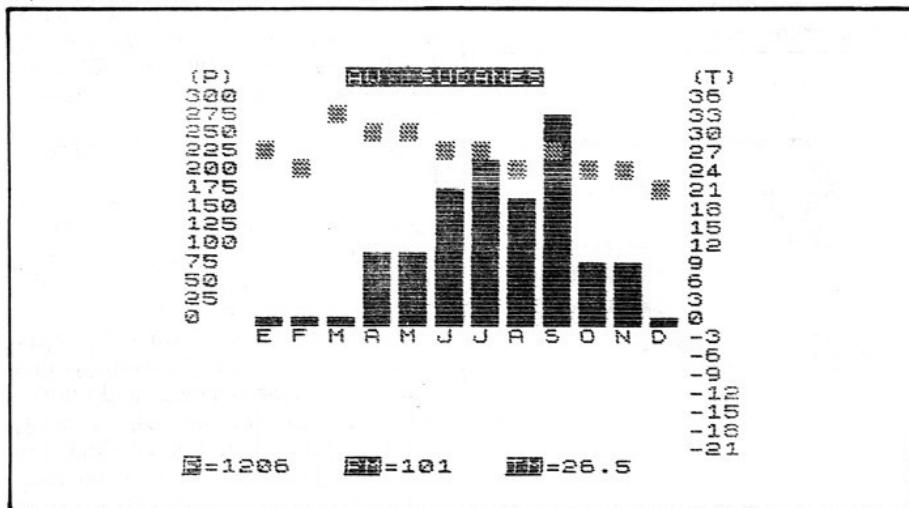
Los conceptos que introducimos ahora son los siguientes. Las variables de cadena constan de una sola letra y el signo dolar (\$). En las líneas 10 y 20 guardamos variables de dos dimensiones, la primera para el número de variables (6 en nuestro caso) y la segunda para la longitud de la palabra, para lo cual nos fijamos en la palabra más larga, NOSOTROS, que tiene ocho letras más el espacio de separación, 9 lugares. Para la variable T\$ solo guardamos 4 letras, que corresponden a la de mayor longitud, AMOS.

El LEN de la línea 170 calcula la longitud del infinitivo del verbo, el número de letras que tiene. La línea 180 calcula la raíz, es decir, toma las letras del verbo V\$ desde la primera hasta (TO) la antepenúltima (L-2).

Evidentemente podríamos ir alargando el programa de forma que calculase los demás tiempos del verbo, e incluso que lo hiciera para las otras dos conjugaciones la 2ª y la 3ª (infinitivos terminados en -ER y en -IR).

Otros Programas

El primer o último cuarto de hora de cada sesión lo dedicábamos a ver otros programas de nuestra biblioteca



particular, marcanos, invasores, ajedrez, dominó y juegos y curiosidades en general, sin olvidar otras aplicaciones serias de la informática como matemáticas, climogramas, archivo de notas, etc., de esta forma el alumno, además de tomar más interés por el asunto asimilando las explicaciones posteriores con más ganas, se daba cuenta de la nueva revolución que se avecinaba, la de la **INFORMATICA**.

Objetivos finales

Estaba claro que esto no era suficiente, el alumno algo había aprendido pero seguramente no sería capaz de confeccionar programas más complejos. Como mucho podría hacer otros que fuesen casi una copia de los anteriores pues en el tintero se nos quedaban conceptos importantes del BASIC sin tocar, como: GOSUB, PRINT AT, PLOT, RND, POKE, etc.

Como había dicho al principio, el aprendizaje del BASIC se presentaba

este curso de forma experimental, de manera que habría que mejorar en el futuro. La única solución sería implantarlo como asignatura. No se trataría de crear una nueva sino de meterla dentro de una ya existente, el E.A.T.P. (Enseñanza de las Actividades Técnico Profesionales), pero dedicando todo el curso a la máquina y al BASIC.

Este objetivo no podría cubrirse con un solo profesor, como en el curso presente, era necesario que otros profesores tomaran conciencia y partido en la nueva situación.

Es por esto por lo que paralelamente al curso de los alumnos también se desarrolló otro para los profesores, en principio de una forma similar. Si bien para la mayoría les sirvió de información, algunos se los tomaron a pecho y con ganas, tanto que incluso adquirieron ZXs para su uso particular. En este caso el aprendizaje del BASIC les quedaba asegurado.

No quisiera terminar sin destacar que estamos confeccionando unos programas, digamos serios, que nos ayuda-

rán en nuestra labor de enseñantes. La mayor parte de matemáticas (sistema de n ecuaciones, producto e inversión de matrices de orden n, ajuste por mínimos cuadrados, errores, etc.) y de otras materias, como la de Historia, que con la colaboración del catedrático de la asignatura, hemos realizado un programa bastante completo sobre *Climogramas*, de forma que el tiempo dedicado a la exposición de este tema podría verse reducido en una décima parte, ya que con solo pulsar una tecla aparece la gráfica *pluviosidad-temperatura* que corresponde a un clima determinado, con lo que, tanto los alumnos como el profesor podrán dedicarse con más detenimiento a los conceptos propios del tema, dejando para el ordenador el trabajo más laborioso.

Por su extensión el propio comentario de estos trabajos con el profesorado tendría que ser objeto de otro artículo.

Jesús GONZALEZ NORES
Profesor Agregado de Física y Química



Cálculo de velocidad de perfusión de fármacos vasoactivos

Es curioso que, si bien algunos hospitales disponen de grandes ordenadores, éstos se emplean fundamentalmente con fines de archivo de Historias Clínicas, o bien para el manejo de problemas muy concretos-control y diagnóstico de arritmias, registros de potenciales evocados, etc-, existiendo a menudo un vacío en la ayuda para problemas muy importantes, aunque sin duda menos complejos, que se resuelven perfectamente con un micro.

En nuestra Unidad, y partiendo de cero, hemos logrado poner a punto en unos meses varios programas, alguno de diagnóstico sofisticado- localización y origen de un estado comatoso- otros, como el que os sometemos hoy, de simple cálculo de dosis de fármacos.

No son programas sofisticados y llenos de trucos, pero realizan bastante bien su función de descargarlos de cálculos rutinarios que antes nos llevaban horas de trabajo, haciéndolos de forma más exacta.

Creemos que ésto puede servir para mostrar otros usos médicos del microordenador, diferentes a la clásicas funciones contables o de fichero.

El tratamiento de los enfermos en una Unidad de Cuidados Intensivos (UCI), precisa con mucha frecuencia del empleo de fármacos muy activos, que se dosifican a menudo en milésimas de miligramo, y que, por ésta misma razón, deben de ser controlados con gran cuidado, empleando para ello bombas de perfusión muy precisas.

Los cambios de dosificación de éstos fármacos son extraordinariamente frecuentes, a tenor de las modificaciones del estado del enfermo. Para evitar tan-

to los repetidos y tediosos cálculos, como los posibles errores de dosificación, algunos fabricantes proporcionan tablas de cálculo, con arreglo al peso del enfermo, y la dosis deseada. En otros casos es el propio personal de la Unidad el que se fabrica sus propias tablas de dosificación. No obstante, estas tablas, cuando existen, tienen el grave inconveniente de emplear una concentración fija de fármaco, por lo que si se emplean dosis importantes de éste, la cantidad de líquido empleada duran-

te el día para vehicularlo puede exceder de lo conveniente para un enfermo adulto, y sobre todo para un niño.

Para solucionar éste problema, hemos puesto a punto en nuestra Unidad un programa de cálculo de fármacos vasoactivos.

En realidad se trata de uno de los subprogramas de un programa general más complejo con el cual realizamos los cálculos diarios de la Unidad. De algunos de éstos subprogramas hablaremos en otra ocasión.

Está escrito para un VIC-20, y como programa aislado cabe cómodamente en la memoria de la Unidad básica. No obstante, al no ser en absoluto un programa sofisticado, puede entrar perfectamente en otros micros. El interés de traerlo aquí es, más que el programa en sí, el mostrar la posibilidad de emplear el microordenador para simplificar el trabajo, y evitar errores en una actividad tan delicada como el tratamiento intensivo de un enfermo.

Tras ser introducidos los datos de filiación del enfermo, aparece en pantalla un índice de subprogramas. Seleccionando el número correspondiente al programa que tratamos, el índice general es substituido por el título del subprograma, al que acompaña otro índice, el de fármacos vasoactivos más usados en la Unidad. La inclusión de cualquier otro que se desee requiere solamente de la adición de los datos correspondientes en las sentencias DATA, así como la modificación de las dimensiones DIM de las listas, y las instrucciones de lectura de éstas.

Seleccionando el número del índice correspondiente al fármaco deseado, aparece en pantalla el nombre de éste; el ordenador pregunta sucesivamente la dosis que se desea calcular en las unidades apropiadas, el tipo de bomba que va a usarse (existen bombas que calculan el número de gotas de una solución que introducen por minuto; otras empujan el émbolo de una jeringa que puede inyectar en general de 0 a 10 ml por hora,

y otras inyectan cantidades muy superiores de líquido por hora), la cantidad de solución que se va a separar, y el máximo diario de líquido que se puede emplear para vehicular el fármaco.

El ordenador muestra entonces la cantidad de medicamento que debe emplearse, la velocidad a que debe programarse cada tipo de bomba, y un factor para cambiar la velocidad si se desea modificar la dosis ante la propia cama del enfermo, sin necesidad de volver a la consola.

El programa da opción a la impresión por un periférico, con o sin los datos de filiación del enfermo, lo cual nos permite incluir el documento de salida de la impresora en la hoja de evolución del enfermo.

Creemos que la disponibilidad de un microordenador poco costoso puede significar una ayuda inestimable para el trabajo cotidiano del siempre atareado personal de una UCI, por ser accesible incluso a los servicios más modestos, reducir a un mínimo los errores de cálculo,

lo, y permitir elaborar fácilmente las variantes juzgadas necesarias para cada Unidad particular.

Doctores Pilar Martínez García
Julián López Alvaro
Juan Carlos Rodríguez Yañez
Ramón Díaz-Alersi Rosety
Nela Sánchez Sánchez
Venus Salcedo Solines
Julián Gil Cebrían

```

100 REM"PROGRAMA PARA EL CALCULO DE VELOCIDAD DE PERFUSION DE FARMACOS VASOACTIV
OS"
110 REM"COPYRIGHT:U.C.I.HOSPITAL CLINICO CADIZ"
120 FORI=1TO9:READD$(I):NEXT:FORI=1TO9:READDD$(I):NEXT:FORI=1TO9:READM$(I):NEXT
130 FORI=1TO9:READML$(I):NEXT:FORI=1TO9:READV$(I):NEXT:FORI=1TO9:READX$(I):NEXT
140 FORI=1TO9:READZ$(I):NEXT
150 DATA DOPAMINA,DOBUTAMINA,NITROPRUSIATO,METAPROTERENOL
160 DATA NORADRENALINA,ADRENALINA,LIDOCAINA 5X,NITROGLICERINA,VASOPRESINA
170 DATA 200,250,50,.5,1,1,500,50 20,MOG/KG/M,MOG/KG/M,MOG/KG/M,MOG/M,MOG/M,MOG/
M
180 DATA MG/M,MOG/M,U.I./M,MG,MG,MG,MG,MG,MG,MG,MG,U.I.,GOTAS/M,ML/H,ML/H,1,3,3
190 DATA1,1,1,1,1,1,1000,1000,1000
200 PRINT"Q":INPUT"HORA ":TI$
210 INPUT"NOMBRE":N$:INPUT"CAMA":C$:INPUT"FECHA":F$:INPUT"PESO":P$:AX
220 REM"PREGUNTA FARMACOS,MODO DE PERFUSION,DOSIS E INTERVALO"
230 PRINT"Q":PRINT:PRINT"F.VASOACTIVOS"
240 PRINT"*****":PRINT
250 FORI=1TO9:PRINTI;"-":DD$(I):NEXT:INPUTR:IFR>3THENAX=1
260 PRINT"MECANISMO DE INFUSION","1)GOTEO " "2)BOMBA VOLUMETRICA","3)JERINGA
PERFUSORA
270 INPUTY:PRINT"DOSIS EN "M$(R):INPUTHH:INPUT"EN QUE VOLUMEN":LL
280 PRINT"CON QUE INTERVALOS EN "M$(R):INPUTO
290 REM"FORMULACION"
300 G=AX*HH*LL*20/(DD(R)*1000):G2=G:IFV=3THENY=2
310 IFR>6THENG2=G*1000
320 GOSUB500
330 G1=(HH+O)*G2/HH-G2
340 REM"PRESENTACION DE RESULTADOS"
350 PRINT"Q":PRINTDD$(R):FORI=1TOLEN(DD$(R)):PRINT"*":NEXT:PRINT
360 PRINT"DOSIFICACION A"HH:M$(R):":PRINTG2*X(Y):V$(Y)
370 IFXX=0THENXX=1
380 PRINT"DISOLVIENDO"XX*DD(R):ML$(R)" EN"LL"ML
390 PRINT"PARA VARIAR LA DOSIS EN"O:M$(R)
400 PRINT"VARIAR LA VELOCIDAD EN"G1*X(Y):V$(Y)
410 GOSUB450
420 REM"REPETICION DE LA PRESENTACION EN CASO DE IMPRESION"
430 GOTO350
440 REM"SUBROUTINA DE IMPRESION RESULTADOS"
450 IFX$="SI"THENX$="":PRINT#1:CLOSE1:END
460 INPUT"DESEA IMPRESION":X$:IFX$<>"SI"THENEND
470 INPUT"CON FILIACION":Y$:IFY$="SI"THENY$="":GOSUB500
480 OPEN1,4:CMD1:RETURN
490 REM"SUBROUTINA IMPRESION DE LA FILIACION"
500 OPEN1,4:CMD1:PRINT"NOMBRE:"N$:PRINT"CAMA:"C$
510 PRINT"FECHA:"F$:PRINT"HORA: ":GOSUB550
520 IFR<4THENPRINT:PRINT"PESO:"P$KG.
530 PRINT#1:CLOSE1:RETURN
540 REM"SUBROUTINA IMPRESION DE HORA (VARIABLE INTERNA DEL VIC)
550 FORT=1TOLEN(TI$)STEP2
560 PRINTMID$(TI$,T,2):IFT<5THENPRINT":":NEXT
570 RETURN
580 REM"SUBROUTINA PARA LIMITACION DE LIQUIDOS EN 24 H."
590 IFM=0THENPRINT"EN QUE VOLUMEN MAX.DE LIQUIDO/DIA":INPUTM:M=M*20/1440
600 IFG2>MTHENG2=M
610 RETURN
620 XX=1+XX:G2=G*Z(R)/XX:GOTO600

```

READY.

Las calculadoras programables también sirven para aprender las tablas de multiplicar

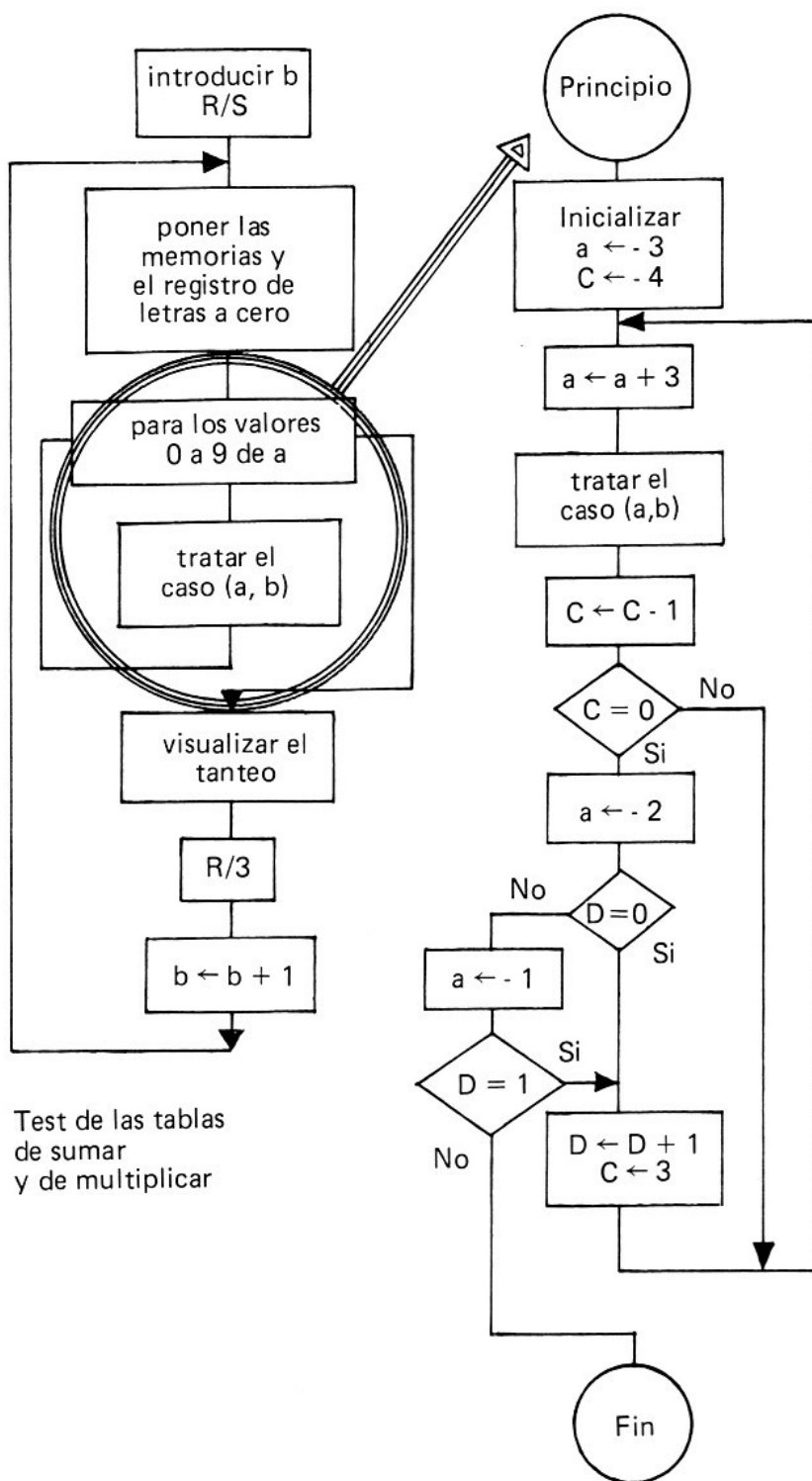
He aquí un pequeño programa de aprendizaje de las tablas de sumar y de multiplicar, escrito para una calculadora programable TI 59. Un niño de 8 años se ha familiarizado rápidamente con este programa, y se ha divertido mucho, a la vez que daba pruebas de gran concentración. Puede que no nos libre del estudio sistemático de las tablas de multiplicar sin las cuales el cálculo numérico no puede llegar a ser eficaz, pero por lo menos, es un principio de aprendizaje sin esfuerzo.

Este programa permite estudiar la tabla del "b".

La pregunta es enunciada de esta forma: a.b y aparece en pantalla. La contestación propuesta se introduce por el teclado y el programa empieza al pulsar la tecla R/S. Si la contestación es correcta, se da una confirmación por visualización de la solución. Si es incorrecta,

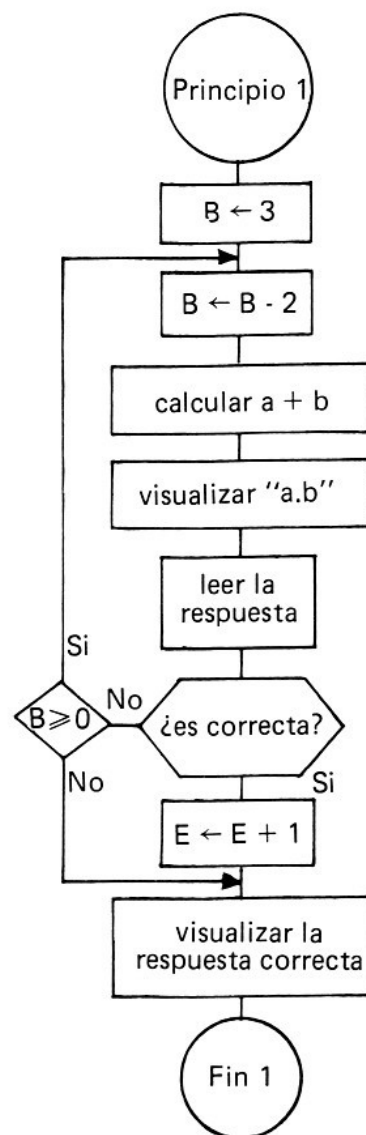
Lista del programa de aprendizaje sobre TI 59

00 41 R/S	18 12 INV	36 94 =	54 35 SUM	77 08 8
01 38 CMs	19 35 SUM	37 41 R/S	55 02 2	78 94 =
02 33 STO	20 03 3	38 74 -	56 34 REC	79 37 x=t
03 09 9	21 34 REC	39 34 REC	57 05 5	80 09 9
04 56 CP	22 08 8	40 05 5	58 49 fix	81 03 3
05 03 3	23 33 STO	41 94 =	59 00 0	82 34 REC
06 93 +-	24 05 5	42 37 x=t	60 59 pause	83 02 2
07 33 STO	25 84 +	43 05 5	61 27 dsz	84 41 R/S
08 08 8	26 34 REC	44 03 3	62 01 1	85 34 REC
09 04 4	27 09 9	45 34 REC	63 02 2	86 09 9
10 33 STO	28 35 SUM	46 03 3	64 02 2	87 84 +
11 00 0	29 05 5	47 47 x>=t	65 93 +-	88 01 1
12 03 3	30 54 :-	48 01 1	66 33 STO	89 94 =
13 33 STO	31 01 1	49 07 7	67 08 8	90 22 GTO
14 03 3	32 00 0	50 22 GTO	68 34 REC	91 00 0
15 35 SUM	33 00 0	51 05 5	69 01 1	92 01 1
16 08 8	34 49 fix	52 06 6	70 37 x=t	93 01 1
17 02 2	35 02 2	53 01 1	71 09 9	94 35 SUM
			72 03 3	95 01 1
			73 84 +	96 03 3
			74 01 1	97 22 GTO
			75 93 +-	98 01 1
			76 33 STO	99 00 0



Test de las tablas de sumar y de multiplicar

"a" toma sucesivamente los valores (-3), 0, 3, 6, 9, (-2), 1, 4, 7, (-2), (-1), 2, 5, 8. Sólo son visualizados los valores de "a" que no están entre paréntesis.



Test: continuación

- B vale 3 y luego 1.
- Si se contesta correctamente la primera vez, B se queda en 1 y E es aumentado de 1.
- Si se contesta mal a la primera, B pasa a -1.
- Si se contesta bien a la segunda vez, E es aumentado de 1.
- Si se contesta mal la segunda vez, como B es negativo, se pasa a la visualización sin aumentar E.
- Para permitir, por ejemplo, 4 contestaciones en vez de 2, poner previamente en B, 7 y no 3.

recta, la pregunta es enunciada una segunda vez. Una segunda contestación falsa ocasiona la visualización de la respuesta correcta y la pregunta es contabilizada como respuesta errónea.

Las "a" se presentan en un suficiente desorden como para impedir que se pueda deducir una respuesta de la precedente.

Una vez agotada la tabla del "b" ("a" varía de 0 a 9), aparece el tanteo (número de respuestas correctas) en pantalla. Al pulsar la tecla R/S, se inicia entonces el estudio de la tabla del "b+1".

Para empezar: pulsar RST, luego R/S, introducir la "b" por la que se quiera comenzar, y hacer aparecer la primera pregunta pulsando R/S.

Nota: La variación de "a" de 0 a 9 y no de 1 a 10 nos es sugerida por una preocupación pedagógica. La experiencia demuestra que la pregunta a x 0 es más difícil que a x 10.

M. Clerc

Programas de 1K Para el ZX-81

TORRES DE HANOI.

¡He aquí una versión gráfica de este clásico juego en justo 1K de memoria!. Vd. tiene tres bases numeradas, y en la base 2 hay una torre de seis pisos. Su misión es trasladar esta torre, piso a piso, a cualquiera de las bases laterales 1 ó 3. Introduzca el número de la base

de origen con N/L y a continuación el número de la base de destino con N/L. Naturalmente no puede poner un piso mayor encima de otro menor, y el ZX81 se encargará de que no lo haga!.

Un buen jugador necesitará más de 60 jugadas para completar la torre. Si lo desea puede usar el modo FAST del ZX81.

ASTEROIDES.

¿Cuántos Kms. puede Vd. recorrer entre la tormenta de asteroides antes de que su nave resulte destruida?. El tamaño de los asteroides varía automáticamente. Mueva su nave a izquierda y derecha con "5" y "8".

TORRES DE HANOI

```

5 LET V=CODE "PI"
10 LET D=V-U
20 LET S=VAL "6"
30 LET T=VAL "3"
40 LET U=T/T
50 DIM Z$(T,S)
53 FOR I=U TO S
55 LET Z$(U+U,I)=STR$ I
59 NEXT I
62 GOTO CODE "Q"
65 INPUT A
80 INPUT B
100 FOR I=U TO S
110 IF Z$(A,I)<>" " THEN GOTO C
ODE "E"
120 NEXT I
130 GOTO U
150 FOR J=U TO I-U
160 IF Z$(B,J)<>" " THEN GOTO U
165 NEXT J

170 LET Z$(A,I)=" "
175 LET Z$(B,I)=STR$ I
180 CLS
182 PRINT "MOV.":D;AT CODE ":",
S+T;"1";TAB CODE "(";"2";TAB COD
E "3"
185 FOR C=U TO T
190 LET M=S+S
195 FOR L=S TO U STEP -U
200 IF Z$(C,L)<>" " THEN GOTO C
ODE "LPRINT"
205 NEXT L
210 NEXT C
215 LET D=D+U
220 GOTO U
225 FOR I=U TO L
290 PRINT AT M,C*(S+U)+I;" ";
300 NEXT I
310 LET M=M-U

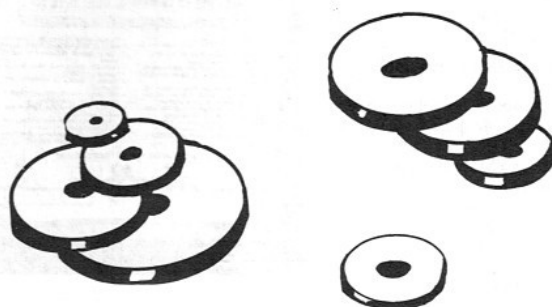
320 GOTO CODE "LN "
```

ASTEROIDES

```

50 LET C=CODE "M"
60 LET D=VAL "2"
70 LET Z=D-D
80 LET L=11
90 LET N=Z
100 LET A$=" "
135 LET X=L
160 LET R=INT (RND*25)
163 IF N=C THEN LET A$=" "
165 IF N=C THEN LET L=L+D
167 IF N=D*C THEN LET A$=" "
168 IF N=D*C THEN LET L=L+D
169 IF N=CODE " " THEN LET L=
L+D
170 PRINT AT 21,R;A$
180 SCROLL
190 SCROLL
255 PRINT AT L-D,X-D;"

260 PRINT AT L,X;
270 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE " " THEN GOTO 500
275 PRINT " "
280 LET X=X+(INKEY$="8" AND X<=
24)-(INKEY$="5" AND X>=D)
290 LET N=N+1
300 GOTO 160
500 PRINT "CRASH";AT Z,Z;"KMS "
;N
```



EL ORDENADOR Y LA HI-FI

Pocos programas han aparecido en las revistas especializadas, referente al tema de alta fidelidad, y muchos menos que puedan ser aplicados al NEWBRAIN A o AD. Este presente programa quiere cortar la tónica general, por lo que les presento un programa que no tan sólo puede ser utilizado por el NEWBRAIN sino que con pocos arreglos puede ser usado con cualquier máquina.

El presente programa ocupa aproximadamente unas 20K de memoria, y con él podemos diseñar filtros divisores de frecuencias para equipos de audio.

Se puede seleccionar en el menú general 7 opciones. Con las 6 primeras podemos definir los filtros de 2 ó 3 vías que más se acerquen a nuestras necesidades. Con la opción 7 podemos calcular la inductancia y el fusible de protección.

Gracias a la cantidad de textos insertos en el programa, este es fácil de usar, incluso para una persona no experta en programación. Además, para aquel que desee modificar el programa, lo tiene muy fácil ya que el programa está separado por REMs, para su mejor acceso.

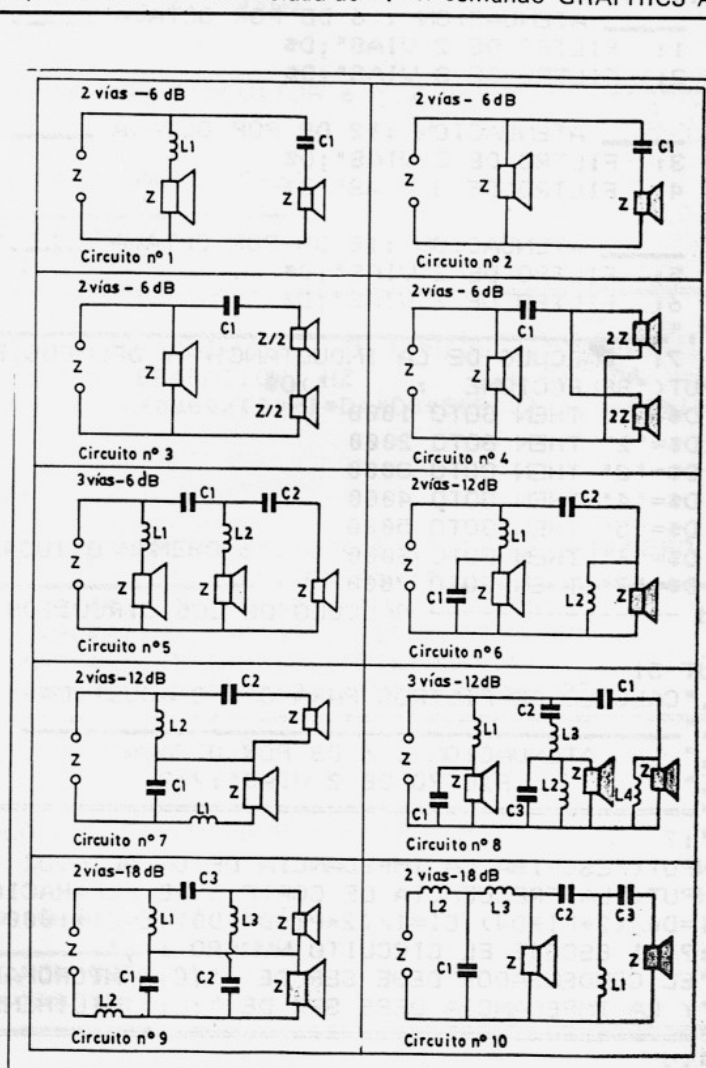
He utilizado el carácter ? en lugar del de PRINT tan sólo por ser más cómodo. Para aquellos que no dispongan del NEWBRAIN tan sólo tienen que cambiar este carácter, y alguna otra cosa. Por ejemplo para tabular dentro de una línea equivale a una TAB (18), y dos, a un TAB (36). La sentencia INPUT (" _texto_ ") variable. Puede ser sustituido en alguna máquina por

un PRINT " _texto_ "; variable: INPUT VARIABLE.

Una de las ventajas del NEWBRAIN es la de poder usar PI sin necesidad de

escribir su valor. Por lo que para otras máquinas se tiene que cambiar.

Para subrayar algunas frases utilizo el comando GRAPHICS A. Y para la



presentación del menú uso GRAPHICS Y.

Su utilización es muy sencilla tan sólo se tiene que ir siguiendo las instrucciones que el mismo programa nos da. Además hay que consultar el esquema que viene como anexo, que representa de forma numerada los posibles esquemas, con su distribución de elementos.

Espero que éste no sea el primer artículo sobre esta nueva máquina, sino que sea el prefacio de otros programas que si la revista lo permite iré publicando.



```

1 REM *****
2 REM ***** PROGRAMA ..... FILTROS PASIVOS DIVISORES DE FRECUENCIAS *
3 REM ***** AUTOR ..... JOSEP PALACIO RIBAS *
4 REM ***** FECHA ..... 24 DE MAIG DE 1.984 *
5 REM ***** (C) ..... J.P.R.(1984) *
6 REM ***** MAQUINA ..... NEWBRAIN AD *
7 REM ***** MEMORIA OCUPADA ... 20689 BYTES *
8 REM *****
90 REM ----- DEFINICION DE VARIABLES Y MENU GENERAL-----
100 PUT 31: CLEAR : OPEN#0,4,"L": DIM D$(50),D0(50),D3(50),D4(50),L1(50),L2(50),C1(
50),C2(50),K$(50),L(50),NE(50),DH(50)
105 ON ERROR GOTO 100
110 ?
120 ?,"----- MENU GENERAL -----":?
130 ?,"----- ATENUACION : 6 DB POR OCTAVA -----":?
135 ?," 1: FILTRO DE 2 VIAS":D$
140 ?," 2: FILTRO DE 3 VIAS":D$
141 ?
145 ?,"----- ATENUACION :12 DB POR OCTAVA -----":?
150 ?," 3: FILTRO DE 2 VIAS":D$
155 ?," 4: FILTRO DE 3 VIAS":D$
156 ?
160 ?,"----- ATENUACION :18 DB POR OCTAVA -----":?
165 ?," 5: FILTRO DE 2 VIAS":D$
170 ?," 6: FILTRO DE 3 VIAS":D$
171 ??:?,"-----":?
175 ?," 7: CALCULO DE LA INDUCTANCIA Y DEL FUSIBLE":?:?
200 INPUT("SELECCIONE : ")D$
201 IF D$="1" THEN GOTO 1000
202 IF D$="2" THEN GOTO 2000
203 IF D$="3" THEN GOTO 3000
204 IF D$="4" THEN GOTO 4000
205 IF D$="5" THEN GOTO 5000
206 IF D$="6" THEN GOTO 6000
207 IF D$="7" THEN GOTO 7000
900 REM ----- CALCULO DE LOS CIRCUITOS 1,2,3 Y 4 -----
-
1000 PUT 31
1010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
1011 ?,"-----"
1015 ?,"----- ATENUACION : 6 DB POR OCTAVA"
1016 ?,"----- FILTRO DE 2 VIAS":?:?
1017 ?,"=====
=====":?
1018 INPUT("ESCRIBA LA IMPEDANCIA DE UN ALTAVOZ , EN OHMIOS : ")D0
1019 INPUT("LA FRECUENCIA DE CORTE O DE SEPARACION , HZ : ")D4
1020 L1=D0/(2*PI*D4):C1=1/(2*PI*D4*D0):L=L1*1000:C=C1*1000000
1030 ??:?"SI ESCOJE EL CIRCUITO NUMERO 1 , "
1035 ?:"EL CONDENSADOR DEBE SER DE " ;C;" MICROFARADIOS"
1050 ?:"Y LA IMPEDANCIA DEBE SER DE " ;L;" MILIHENRIOS"
1060 ?,"=====
=====":?

```



```

1065 INPUT("FRECUENCIA DE TRANSFERENCIA, ALTAVOZ DE AGUDOS, HZ : ")D3
1070 DD=D0/4:CC=1000000/(2*PI*D3*DD)
1080 ?:"PERO SI ESCOJE UNO DE LOS CIRCUITOS 2,3 Y 4,"
1090 ?"ENTONCES EL CONDENSADOR SERA DE ";CC;" MICROFARACIOS"
1100 ?"=====
=====":INPUT("Si desea continuar pulse S ")M$:IF M$="S" THEN GOTO 2000
1101 IF M$<>"S" THEN GOTO 100
1900 REM ----- CALCULO DEL CIRCUITO 5 -----
--
2000 PUT 31
2010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
2011 ?,"-----"
2015 ?,"          ATENUACION : 6 DB POR OCTAVA"
2016 ?,"          FILTRO DE 3 VIAS":?:?
2017 ?"=====
=====":?
2018 INPUT("ESCRIBA LA IMPEDANCIA DE UN ALTAVOZ , EN OHMIOS : ")D0
2019 INPUT("FRECUENCIA DE TRANSFERENCIA, ALTAVOZ DE GRAVES, HZ : ")D1
2020 INPUT("LA FRECUENCIA DE CORTE O DE SEPARACION , HZ : ")D4
2021 L1=D0*1000/(2*PI*D1):L2=D0*1000/(2*PI*D3)
2030 C1=1000000/(2*PI*D1*D0):C2=1000000/(2*PI*D3*D0)
2040 ?"=====
=====":?,"          VEA EL CIRCUITO NUMERO 5"
2041 ?,"          "
2050 ?"EN SERIE CON ALTAVOZ DE GRAVES, L1: ";L1;" MILIENRIOS"
2060 ?"EN SERIE CON ALTAVOZ DE MEDIOS, C1: ";C1;" MICROFARADIOS"
2070 ?"          Y L2: ";L2;" MILIENRIOS"
2080 ?"EN SERIE CON ALTAVOZ DE AGUDOS, C2: ";C2;" MICROFARADIOS"
2100 ?"=====
=====":INPUT("Si desea continuar pulse S ")M$
2101 IF M$<>"S" THEN GOTO 100
2102 IF M$="S" THEN GOTO 3000
2900 REM ----- CALCULO DE LOS CIRCUITOS 6 Y 7 -----
--
3000 PUT 31
3010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
3011 ?,"-----"
3015 ?,"          ATENUACION :12 DB POR OCTAVA"
3016 ?,"          FILTRO DE 2 VIAS":?:?
3017 ?"=====
=====":?
3018 INPUT("ESCRIBA LA IMPEDANCIA DE UN ALTAVOZ , EN OHMIOS : ")D0
3019 INPUT("LA FRECUENCIA DE CORTE O SEPARACION, HZ : ") D4
3020 L1=D0*SQR(2)*1000/(2*PI*D4):C1=1000000/(2*PI*D4*D0*SQR(2)):LA=D0*1000/(2*PI
*D4)
3030 LB=D0*1000/(1.6*2*PI*D4)
3070 CA=1.6*1000000/(2*PI*D4*D0)
3080 CB=1000000/(2*PI*D4*D0)
3090 ?"COMPONENTE";TAB(18);"CIRCUITO NUMERO 6";TAB(45);"CIRCUITO NUMERO 7"
3091 ?"          ";TAB(18);"          ";TAB(45);"          "
3100 ? "L1 VALE : ";TAB(21);L1;TAB(48);LA;TAB(63); "MILIENRIOS"
3110 ? "C1 VALE : ";TAB(21);C1;TAB(48);CA;TAB(63);"MICROFARADIOS"
3120 ? "L2 VALE : ";TAB(21);L1;TAB(48);LB;TAB(63);"MILIHENRIOS"
3130 ? "C2 VALE : ";TAB(21);C1;TAB(48);CB;TAB(63); "MICROFARADIOS"
3140 ?:"=====
=====":?:INPUT("Si desea continuar pulse S ")M$
3141 IF M$<>"S" THEN GOTO 100
3142 IF M$="S" THEN GOTO 4000
3900 REM ----- CALCULO DEL CIRCUITO 8 -----
--
4000 PUT 31
4010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
4011 ?,"-----"
4015 ?,"          ATENUACION :12 DB POR OCTAVA"
4016 ?,"          FILTRO DE 3 VIAS":?:?
4017 ?"=====
=====":?

```

```

4018 INPUT("ESCRIBA LA IMPEDANCIA DE UN ALTAVOZ , EN OHMIOS : ")D0
4019 INPUT("FRECUENCIA DE TRANSFERENCIA, ALTAVOZ DE GRAVES, HZ: ") D1
4020 INPUT("FRECUENCIA DE TRANSFERENCIA, ALTAVOZ DE AGUDOS, HZ: ") D3
4021 L1=D0*SQR(2)*1000/(2*PI*D1)
4022 L2=D0*SQR(2)*1000/(2*PI*D1)
4030 L3=D0*SQR(2)*1000/(2*PI*D3)
4031 L4=D0*SQR(2)*1000/(2*PI*D3)
4050 C1=1000000/(2*PI*D0*D1*SQR(2))
4051 C2=1000000/(2*PI*D0*D1*SQR(2))
4060 C3=1000000/(2*PI*D0*D3*SQR(2))
4061 C4=1000000/(2*PI*D0*D3*SQR(2))
4080 ?,"      CIRCUITO NUMERO 8"
4081 ?,"      "
4100 ?"L1:";L1;" MILIHENRIOS";?"      C1:";C1;" MICROFARADIOS"
4110 ?"L2:";L2;" MILIHENRIOS";?"      C2:";C2;" MICROFARADIOS"
4120 ?"L3:";L3;" MILIHENRIOS";?"      C3:";C3;" MICROFARADIOS"
4130 ?"L4:";L4;" MILIHENRIOS";?"      C4:";C4;" MICROFARADIOS"
4140 ?:"=====
=====":?:INPUT("Si desea continuar pulse S ")M$
4141 IF M$<>"S" THEN GOTO 100
4142 IF M$="S" THEN GOTO 5000
4900 REM ----- CALCULO DEL CIRCUITO 9 -----
--
5000 PUT 31
5010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
5011 ?,"      "
5015 ?,"      ATENUACION :18 DB POR OCTAVA"
5016 ?,"      FILTRO DE 2 VIAS":?:?
5017 ?:"=====
=====":?
5018 INPUT("ESCRIBA LA IMPEDANCIA DE UN ALTAVOZ , EN OHMIOS : ")D0
5019 INPUT("LA FRECUENCIA DE CORTE O SEPARACION, HZ : ") D4
5020 L1=D0*1000/(3*PI*D4);LX=D0*1000/(4*PI*D4)
5030 L2=2*D0*1000/(2*PI*D4);LB=L2
5040 L3=D0*1000/(2*PI*D4);LC=D0*1000/(3.2*PI*D4)
5050 C1=1600000/(2*PI*D4*D0);CA=2000000/(2.0*PI*D0*D4)
5060 C2=1000000/(2*PI*D4*D0);CB=1000000/(3.2*PI*D0*D4)
5070 C3=1000000/(4*PI*D4*D0);CC=1000000/(2.0*PI*D0*D4)
5075 LA=D0*1000/(4*PI*D0)
5110 ?:"=====
=====":?
5111 ? TAB(13);"CIRCUITO NUMERO 9";TAB(41);"CIRCUITO NUMERO 10"
5112 ? TAB(13);"      ";TAB(41);"      "
5120 ?"L1= ";TAB(15);L1;TAB(43);LX;TAB(60);"MILIHENRIOS"
5130 ?"L2= ";TAB(15);L2;TAB(43);LB;TAB(60);"MILIHENRIOS"
5140 ?"L3= ";TAB(15);L3;TAB(43);LC;TAB(60);"MILIHENRIOS"
5150 ?"C1= ";TAB(15);C1;TAB(43);CA;TAB(60);"MICROFARADIOS"
5160 ?"C2= ";TAB(15);C2;TAB(43);CB;TAB(60);"MICROFARADIOS"
5170 ?"C3= ";TAB(15);C3;TAB(43);CC;TAB(60);"MICROFARADIOS"
5180 ?:"=====
=====":?:INPUT("Si desea continuar pulse S ")M$
5181 IF M$<>"S" THEN GOTO 100
5182 IF M$="S" THEN GOTO 6000
5900 REM ----- INHIBICION DEL CIRCUITO 10 -----
--
6000 PUT 31
6010 ?,"CALCULO DE FILTROS PARA CAJAS ACUSTICAS"
6011 ?,"      "
6015 ?,"      ATENUACION :18 DB POR OCTAVA"
6016 ?,"      FILTRO DE 3 VIAS":?:?
6017 ?:"=====
=====":?
6030 ?".....NO LE RECOMENDAMOS QUE EMPLEE ESTE TIPO DE CIRCUITOS PAS
IVOS":?,"      PARA CONFIGURAR ESTE TIPO DE FILTRO"
6037 ?:"=====
=====":?

```



```

6040 FOR Z=0 TO 1600 :NEXT Z:GOTO 6050
6050 ?:"=====
=====":?:INPUT("Si desea continuar pulse S ")M$
6051 IF M$<>"S" THEN GOTO 100
6522 IF M$="S" THEN GOTO 7000
6900 REM ----- CALCULO DE LA INDUCTANCIA Y DEL FUSIBLE -----
--
7000 PUT 31: TAB(20); "CONSTRUCCION ELEMENTAL DE UNA BOBINA DE FILTRO"
7001 ? TAB(20): " :?:?

7010 ?" SI NOS PROPORCIONA ALGUNOS DATOS , LE DIREMOS LA BOBINA QUE DEBE"
7020 ?" CONSTRUIR PARA CONSEGUIR UNA CIERTA IMPEDANCIA APROXIMADA"
7030 ?" ES PRECISO EMPLEAR HILO DE COBRE ESMALTADO DEL DIAMETRO INDICADO"

7031 ?:"=====
=====
7040 ?" SI LA BOBINA NO SATISFACE SUS NECESIDADES FISICAS DE ESPACIO O"
7050 ?" DE RESISTENCIA OHMICA, VAYA ALTERANDO LOS VALORES SOLICITADOS "
7060 ?" HASTA CONSEGUIR LA QUE MAS LE CONVENGA"
7070 ?:"=====
=====
7075 ? TAB(15); "CUANDO ESTE PREPARADO PULSE <NEWLINE>"; K$;:INPUT(" ")K$:GOTO 710
0
7100 PUT 31: TAB(20); "CONSTRUCCION ELEMENTAL DE UNA BOBINA DE FILTRO"
7101 ? TAB(20); " :?:?
7102 ?" ESCRIBA LOS SIGUIENTES DATOS"
7103 ?"=====
7110 INPUT("EL VALOR DE LA INDUCTANCIA 'L' EN MILIHENRIOS: ")L
7120 INPUT("EL DIAMETRO DEL NUCLEO DE LA BOBINA EN MM : ")A1:A=A1/10
7130 INPUT("LA POTENCIA DE UN CANAL EN WATIOS 'RMS' : ")W
7140 INPUT("LA IMPEDANCIA DE LA CAJA, 4 , 8 0 16 OHMIOS : ")IM
7141 IF IM<>4 GOTO 7165
7160 K=2:GOTO 7200
7165 IF IM=8 GOTO 7170
7166 IF IM<>8 GOTO 7175
7170 K=1:GOTO 7200
7175 IF IM=16 GOTO 7180
7180 K=0.5:GOTO 7200
7200 NE=SQR((L*1000*25*A)/(0.315*A^2))
7210 DH=0.5*A1/SQR(NE)
7220 LH=NE*2*PI*1.5*A/100
7230 RO=(0.0172*4*LH)/(PI*DH^2)
7290 ?:"=====
=====
7300 ?" PARA CONSTRUIR LA BOBINA OPTIMA DE ";L;" MILIHENRIOS"
7310 ?" DEBE DE TOMAR UNOS ";LH;" METROS"; " DE HILO DE ";DH;" MM."; " DE DIAMETRO"
7320 ?" BOBINAR ";NE;" ESPIRAS "; " SOBRE UN NUCLEO DE ";A1;" MM."; " DE DIAMETRO"
7321 ?:"=====
=====
7330 ?" LONGITUD DEL BOBINADO: ";A1/2;" MM. ALTURA DEL MISMO: ";A1/2;" MM."
7331 ?:"=====
=====
7340 ?" LA RESISTENCIA TOTAL DE ESTA BOBINA SERA DE ";RO;" OHMIOS":IF RO>0.99 THE
N PRINT "QUE ES DEMASIADA PARA ESTE USO. DEBE DE CONFIGURAR OTRA BOBINA,":?:"=====
=====
7341 IF W<80 GOTO 7345
7342 IF W>80 OR W<100 GOTO 7346
7345 FA=0.04*K*W:GOTO 7350
7346 FA=3*K:GOTO 7350
7350 ?" DEBE COLOCAR UN FUSIBLE DE ";FA;" AMPERIOS EN LA CAJA ACUSTICA."
7360 ?:"=====
=====
=====":?:INPUT("Si desea continuar pulse S ")M$
7361 IF M$<>"S" THEN GOTO 100
7362 IF M$="S" THEN GOTO 8000
8000 PUT 31:?:?:?:?:?:?:?:?:?:?:
8100 ?,"FIN DEL PROGRAMA":?:?:?
8200 ?,"Para regresar al menu principal pulse <NEWLINE> ";M$;:INPUT(" ")M$:GOTO
100
8300 END

```

Espíritu ¿estás aquí? o los fantasmas del Commodore 64 (1.ª Parte)

Esta es una historia para no dormir que os transportará a través de la noche de los tiempos, a la búsqueda de los primeros neofantasmas. No abandonéis, materializarlos en la pantalla de vuestro Commodore 64, en toda su dimensión e incluso en color gracias a este programa Basic.

Un fantasma es esencialmente un bloque gráfico definido por el utilizador en la memoria, y controlado por un procesador 6566, circuito de vídeo del Commodore 64.

El programador elegirá la forma, los desplazamientos y el color; y podrá controlarlos fácilmente mediante POKes hábilmente elegidos.

Haced pasar los fantasmas por debajo o por encima de todo lo que se encuentra en la pantalla, sus eventuales colisiones serán detectadas, pudiendo con ello realizar fantásticas ilusiones de relieve.

Pero, veamos los problemas técnicos antes de que nuestra imaginación se desborde.

Cada fantasma está constituido por una matriz de veinticuatro puntos de longitud por veintidós puntos de altura. Cada punto representa una unidad elemental de la pantalla, es decir ciento veinticuatro puntos por fantasma.

*Por favor, dibújame
cualquier cosa
fantástica*

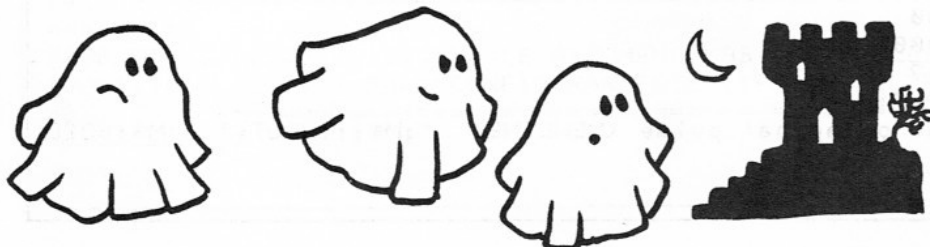
La primera dificultad reside en la realización del dibujo del fantasma, además de la asignación de los valores numéricos necesarios para la escritura en memoria. Este trabajo puede hacerse en una hoja de papel, trazando

una cuadrícula de veintidós líneas por veinticuatro columnas (diagrama 1 de la página siguiente).

Procediendo de igual forma con los tres octetos de cada línea del diseño, obtendremos los 63 (21 x 3) números a introducir en RAM.

Si juzgáis pesado este trabajo, utilizad el programa «generador de fantasmas».

A mí, las historias de fantasmas me dan miedo.



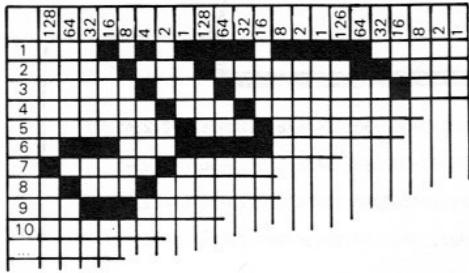


Diagrama 1

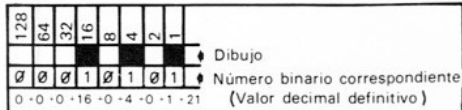


Diagrama 2

ADR.	+	CONTENIDO
53248	0	Coord. X del fantasma Ø
53249	1	coord. Y du fantome 0
53250	2	— X — 1
53251	3	— Y — 1
53252	4	— X — 2
53253	5	— Y — 2
53254	6	— X — 3
53255	7	— Y — 3
53256	8	— X — 4
53257	9	— Y — 4
53258	10	— X — 5
53259	11	— Y — 5
53260	12	— X — 6
53261	13	— Y — 6
53262	14	— X — 7
53263	15	— Y — 7
53264	16	Coord. X de mayor peso
53269	21	Selección de los fantasmas
53271	23	Ampliación vertical de los fantasmas
53277	29	Ampliación horizontal de los fantasmas
53287	39	Color del fantasma Nº Ø
53286	40	— 1
53289	41	— 2
53290	42	— 3
53291	43	— 4
53292	44	— 5
53293	45	— 6
53294	46	— 7

Tabla 3

128	64	32	16	8	4	2	1
0	0	0	0	1	0	0	0
7	6	5	4	3	2	1	0

Tabla 4

128	64	32	16	8	4	2	1
0	0	0	0	1	0	0	0
7	6	5	4	3	2	1	0

Tabla 5



Dejad a un lado, de momento, vuestro fantasma numérico para considerar desde más cerca el famoso 6566. Este procesador utiliza un cierto número de direcciones (la primera comienza en 53248) de vital importancia (bromas aparte) para nuestros fantasmas (tabla 3 de al lado).

Un segundo registro, que va de las direcciones 2040 a 2047, nos resultará de utilidad inmediata.

Dirección	Contenido
2040	nº de bloque del fantasma 0
2041	nº de bloque del fantasma 1
2042	nº de bloque del fantasma 2
2043	nº de bloque del fantasma 3
2044	nº de bloque del fantasma 4
2045	nº de bloque del fantasma 5
2046	nº de bloque del fantasma 6
2047	nº de bloque del fantasma 7

Con la lectura de estas tablas habreis podido constatar que pueden generarse ocho fantasmas simultaneamente. En realidad, es posible aumentar este número, pero no es este nuestro propósito de hoy.

Fantasmas en color y en toda su dimensión

He aquí, paso a paso, las instrucciones a introducir desde teclado para provocar el naci-

miento de un fantasma de lo más fascinante; referiros cada vez al programa de demostración.

Líneas 10 a 20: diseño. Colocad en DATA los sesenta y tres valores numéricos de vuestro fantasma obtenidos gracias al método descrito anteriormente.

Línea 30: dirección de partida. La dirección 53248 es la dirección de base del 6566, a partir de ella se calculan la mayor parte de las direcciones útiles a los fantasmas.

Línea 40: validación de fantasmas. Podremos seleccionar, por ejemplo, los fantasmas 3 y 5 de entre los ocho posibles con POKE en 53269 (tabla 4 de al lado).

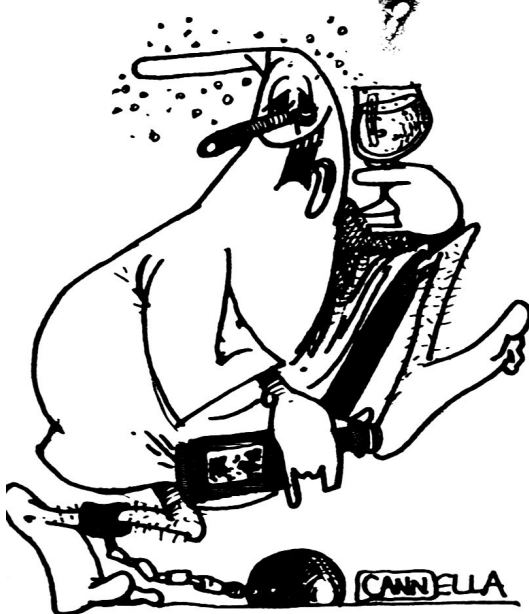
Un fantasma validado se traduce en un 1 en el octeto (que transformaremos al valor decimal correspondiente) contenido en esta dirección.

Líneas 50 y 60: bloques de memoria. Utilizaremos tres bloques para memorizar el dibujo de nuestro fantasma:

nº de bloque	direcciones
13	832 a 894
14	896 a 958
15	960 a 1022

Hacemos saber a los fantasmas 3 y 5 que tienen memorizado su diseño en el bloque número 13 mediante dos POKES (tabla 3 de al lado).

3 siglos habitando una destilería, eso deja marcas hips hips...



Línea 70: transferencia del diseño. Transferimos nuestros sesenta y tres DATA al bloque 13.

Línea 80: color. Nuestros dos fantasmas tendrán el color nº 13 (verde)... ¡¡¡ obligatoriamente!!!

Líneas 90 y 100: tamaño. Deseamos un fantasma nº 3 de talla grande. Serán necesarios dos POKE. Uno para aumentar la altura y el otro para duplicar la anchura. El valor del POKE se obtiene de la forma habitual (tabla 5 de la página anterior).

Líneas 110 y 120: posición de la pantalla: Dos POKEs más son necesarios para indicar a cada fantasma la posición que deberá ocupar en la pantalla. Uno de ellos para indicar la posición horizontal, y el otro para la vertical. Si cambiáis estos valores, en un bucle por ejemplo, vuestro fantasma se desplazará por la pantalla.

Es al final cuando comienzan los problemas

¡Horror! Después de la verificación del programa y a continuación del RUN, podréis sentir vuestros crispados cabellos. Vuestra máquina se encuentra embrujada y, peor aún, ningún borrado de pantalla podrá ahuyentar a los ecotoplasmas, solamente con las teclas SPOP/RESTORE podréis liberaros de esta visión...

Juan-Pedro Lalevé

```
1 REM12345678901234567890123456789012345678901234567890123456789012345
2 REM12345678901234567890123456789012345678901234567890123456789012345
3 REM12345678901234567890123456789012345678901234567890123456789012345
4 REM12345678901234567890123456789012345678901234567890123456789012345
```

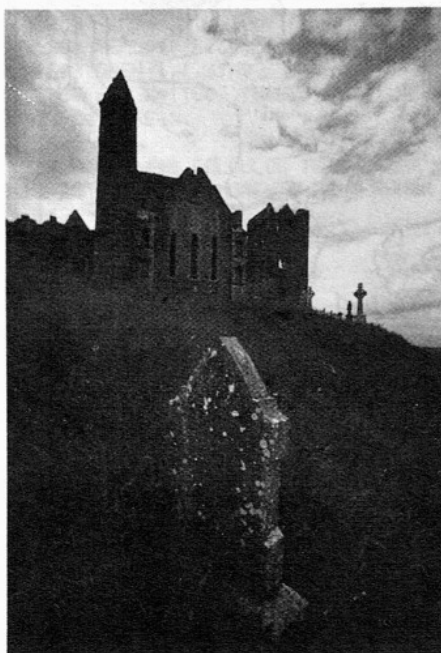
```
5 :
6 REM (C) COPYRIGHT EL ORDENADOR PERSONAL
10 PRINT "D" TAB(8) "GENERADOR DE FANTASMAS"
15 FOR X=55296 TO 56295: POKE X, 3: NEXT
20 :
25 PRINT "X0001=ARRIBA"
30 PRINT "X0002=ABAJO"
35 PRINT "X0003=IZQUIERDA"
40 PRINT "X0004=DERECHA"
45 PRINT "X0005=SHIFT=TRAZA"
50 PRINT "X0006=TERMINAR"
55 PRINT "X0007=CALCULAR"
60 PRINT "X0008=ANULAR"
65 PRINT "X0009=PEQUEÑO"
70 PRINT "X0010=GRANDE"
75 PRINT "X0011=MODO DIBUJO"
80 DIM H(62): AB=53248: L=1078: W=55350
85 FOR X=0 TO 25: POKE L+X, 102: POKE W+X, 1
90 POKE L+22*40+X, 102: POKE W+22*40+X, 1: NEXT
95 FOR X=0 TO 22: POKE L+X*40, 102: POKE W+X*40, 1
100 POKE L+25*X*40, 102: POKE W+25*X*40, 1: NEXT
105 P=L+41
110 POKE P, 67
115 POKE 198, 0: WAIT 198, 1
120 A=PEEK(197)
125 IF PEEK(653) THEN POKE P, 160: GOTO 135
130 POKE P, 32
135 IF A=33 THEN M=-40: GOTO 175
140 IF A=34 THEN M=-1: GOTO 175
145 IF A=37 THEN M=1: GOTO 175
150 IF A=36 THEN M=40: GOTO 175
155 IF A=54 THEN POKE AB+21, 0: RUN
160 IF A=55 THEN 185
165 IF A=57 THEN 225
170 GOTO 110
175 IF PEEK(P+M)=102 THEN 110
180 P=P+M: GOTO 110
185 PRINT "COMPILE"
190 N=0: FOR Y=1 TO 21: FOR X=0 TO 2: FOR Z=7 TO 0 STEP -1
195 IF PEEK(L+X*8+Y*40+Z)=160 THEN C=C+2: Z
200 NEXT Z: POKE 832+N, C: H(N)=C: N=N+1: C=0
205 NEXT X, Y
210 POKE AB+21, 12: POKE 2042, 13: POKE 2043, 13: POKE AB+4, 90: POKE AB+5, 150
215 POKE AB+6, 80: POKE AB+7, 180: POKE AB+23, 8: POKE AB+29, 8
220 PRINT "MODO DIBUJO": GOTO 110
225 Z=0: K=PEEK(43)+PEEK(44)*256+4: POKE AB+21, 0
230 PRINT "TERMINA(S/N)?" : POKE 198, 0
235 GET A: IF A=C: "N" OR A=C: "S" THEN 235
240 IF A="N" THEN 220
245 PRINT "PASANDO EL DIBUJO A (DATA)": PRINT "UN MOMENTO POR FAVOR..."
250 POKE K, 131: J=1
255 I$=STR$(H(Z)): FOR X=2 TO LEN(I$)
260 POKE K+J, ASC(MID$(I$, X, 1)): J=J+1
265 NEXT X
270 Z=Z+1: IF Z=63 THEN 285
275 IF J<=71 THEN POKE K+J, 44: J=J+1: GOTO 255
280 FOR X=J TO 75: POKE K+X, 32: NEXT: POKE K+76, 0: K=K+81: GOTO 250
285 FOR X=J TO 75: POKE K+X, 32: NEXT: FOR X=76 TO 78: POKE K+X, 0: NEXT
290 LIST: END
```

```
5 REM DEMOSTRACION FANTASMAGORICA DE LOS RESULTADOS
6 REM (C) COPYRIGHT EL ORDENADOR PERSONAL
10 DATA 0,60,0,1,255,128,7,255,224,15,254,240,15,248,248,31,128,120,31,0
15 DATA 120,28,0,56,24,0,24,25,199,16,8,0,16,8,0,16,4,48,32,4,0,32,2,4
20 DATA 64,2,56,128,1,1,128,0,130,128,0,124,64,1,0,240,255,129,252
30 AB=53248
40 POKE AB+21,40: REM VALIDACION
50 POKE 2043,13: REM DIRECCION DE DESTINO
60 POKE 2045,13
70 FOR Q=0 TO 62: READ DA: POKE 832+Q, DA: NEXT Q: REM ESCRITURA DEL DIBUJO
80 POKE AB+42,13: POKE AB+44,13: REM COLOR
90 POKE AB+23,8: REM TAMAÑO
100 POKE AB+29,8
110 POKE AB+6,150: POKE AB+7,180: REM COORDENADAS
120 POKE AB+10,90: FOR Y=0 TO 200: POKE AB+11,Y: NEXT
130 END
```


Espíritu, ¿estás aquí? o los fantasmas del commodore 64 (2ª parte)



En la primera parte de este artículo, habéis podido conocer algunos espectros abominables, creados en la sombra de vuestro Commodore 64. Esta segunda parte os permitirá verlos a todo color. También en esta ocasión, terror y escalofríos están presentes en el programa.



Hasta ahora hemos manipulado ectoplasmas monocromáticos. Hagamos surgir nuevos monstruos, artísticamente más decorados, al precio de una mayor complejidad técnica: fantasmas multicolores compuestos de tres colores diferentes a la vez (cuatro teniendo en cuenta el color de fondo que siempre es transparente).

Aunque el diseño de un fantasma multicolor también esté codificado a partir de una matriz de 21 líneas por 24 columnas, la codificación es diferente a la anterior. En lugar de un bit para representar un punto apagado o encendido, se utilizan grupos de dos bits que definen tres elementos:

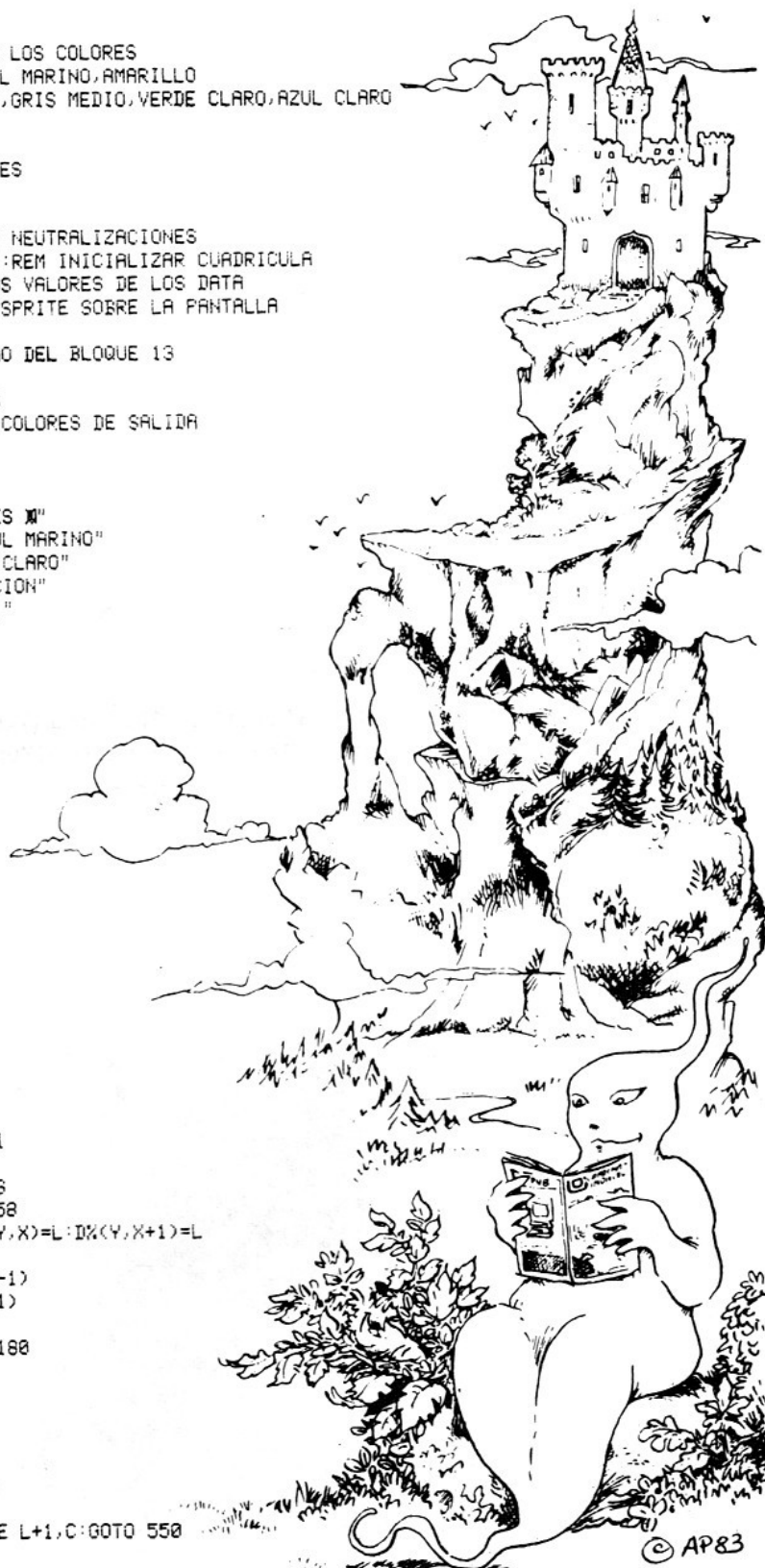
- * color de base del fantasma,
- * color suplementario número 1,
- * color suplementario número 2.

Un octeto formado de cuatro grupos de dos bits, determina la posición de los puntos eventualmente encendidos.

Es evidente que esta codificación en dos bits proporcionará una menor definición de imagen, puesto que serán encendidos o apagados dos puntos a la vez.

READY.

```
1 REM*****
2 REM
3 REM  EDITOR DE SPRITES MULTICOLORES
4 REM
5 REM  (C) COPYRIGHT EL O.P.
6 REM
7 REM*****
100 :
110 REM ** INICIALIZACIONES **
120 :
130 DIM D%(21,24),V(63),C$(15)
140 FOR Q=0 TO 15:PEEK C$(Q):NEXT:REM NOMBRES DE LOS COLORES
150 DATA NEGRO,BLANCO,ROJO,CYAN,ROSA,VERDE,AZUL MARINO,AMARILLO
160 DATA NARANJA,MARRON,ROJO CLARO,GRIS FUERTE,GRIS MEDIO,VERDE CLARO,AZUL CLARO
165 DATA GRIS CLARO
170 :
180 E=1039:F=E+54272:REM DIR. PANTALLA / COLORES
190 PRINT"UN MOMENTO POR FAVOR ..."
200 AB=53248:REM DIR. DE BASE DE LOS SPRITES
210 POKE AB+21,0:POKE AB+23,0:POKE AB+29,0:REM NEUTRALIZACIONES
220 FOR L=1 TO 21:FOR C=1 TO 24:D%(L,C)=58:NEXT C:L:REM INICIALIZAR CUADRICULA
230 FOR Q=1 TO 63:V(Q)=0:NEXT:REM INICIALIZAR LOS VALORES DE LOS DATA
240 POKE AB,60:POKE AB+1,200:REM POSICION DEL SPRITE SOBRE LA PANTALLA
250 POKE 2040,13:REM SPRITE 0 EN EL BLOQUE 13
260 FOR Q=1 TO 63:POKE AB+Q,V(Q):NEXT:REM RELLENO DEL BLOQUE 13
270 POKE AB+21,1:REM VALIDA EL SPRITE 0
280 POKE AB+28,1:REM MULTI-COLOREA EL SPRITE 0
290 POKE AB+37,0:POKE AB+38,1:POKE AB+39,14:REM COLORES DE SALIDA
300 :
310 REM *** PRESENTACION DE LA PANTALLA ***
320 :
330 PRINT"  FANTASMAS  ":PRINT"  MULTICOLORES  "
340 PRINT"  DEL  BORRADO  ":PRINT"  FONDO  AZUL MARINO  "
350 PRINT"  1  0  NEGRO  ":PRINT"  2  1  AZUL CLARO  "
360 PRINT"  3  2  BLANCO  ":PRINT"  4  3  COMPILACION  "
370 PRINT"  5  4  EXP. HOR.  ":PRINT"  6  5  EXP. VER.  "
380 PRINT"  7  6  CALCULA DATAS  "
390 PRINT"  8  7  ANULA  ":PRINT"  9  8  SALVA K7  "
400 PRINT"  0  9  CARGA K7  ":PRINT"  1  0  TERMINA  "
410 :
420 REM *** DIBUJO DE LA CUADRICULA ***
430 Y=0
440 FOR L=1 TO 21:FOR C=1 TO 24
450 Y=Y+1
460 POKE E+Y,D%(L,C):POKE F+Y,12
470 NEXT C
480 Y=Y+16
490 NEXT L
500 X=1:Y=1
510 GOTO 750
520 :
530 REM *** ANALISIS DE LAS PETICIONES ***
540 :
550 POKE 198,0:WAIT 198,1:GET A$
560 L=E+X+(Y-1)*40:C=D%(Y,X)
570 POKE L,C:POKE L+1,C
580 :
590 IF A$=CHR$(17) THEN Y=Y+1:IF Y>21 THEN Y=1
600 IF A$=CHR$(145) THEN Y=Y-1:IF Y< 1 THEN Y=21
610 IF A$=CHR$(29) THEN X=X+2:IF X>23 THEN X=1
620 IF A$=CHR$(157) THEN X=X-2:IF X< 1 THEN X=23
630 IF A$=CHR$(20) THEN D%(Y,X)=56:D%(Y,X+1)=58
640 IF A$>"0" AND A$<"4" THEN L=48+VAL(A$):D%(Y,X)=L:D%(Y,X+1)=L
650 IF A$="" THEN 790
660 IF A$="H" THEN POKE AB+29,ABS(PEEK(AB+29)-1)
670 IF A$="V" THEN POKE AB+23,ABS(PEEK(AB+23)-1)
680 IF A$="D" THEN 980
690 IF A$="C" OR A$="S" OR A$="A" OR A$="T" THEN 1180
700 IF A$=CHR$(133) THEN L=33:GOSUB 1070
710 IF A$=CHR$(134) THEN L=37:GOSUB 1070
720 IF A$=CHR$(135) THEN L=39:GOSUB 1070
730 IF A$=CHR$(136) THEN L=38:GOSUB 1070
740 :
750 REM *** TRATAMIENTO DE LAS PETICIONES ***
760 :
770 L=E+X+(Y-1)*40:C=D%(Y,X)+128:POKE L,C:POKE L+1,C:GOTO 550
780 :
790 REM *** COMPILACION ***
```



© AP83


```

800 Y=0
810 FOR L=1TO21:FOR X=0TO2
820 Y=Y+1:V(Y)=0
830 FOR C=1TO7STEP 2
840 Q=D%(L,X*8+C)-48
850 IF Q<0OR Q>3 THEN Q=0
860 V(Y)=V(Y)+2*(7-C)*Q
870 NEXT C
880 POKE 831+Y,V(Y)
890 NEXT X,L
900 GOTO 500
910 :
920 REM *** VERIFICACION DE LA PETICION ***
930 PRINT "XXXXXXXXXXXXXXXXXXXXA$":EJECUCION?"
940 POKE 198,0:WAIT 198,1:GET N$
950 PRINT "J"
960 RETURN
970 :
980 REM *** GENERACION DE DATAS ***
990 PRINT "J"
1000 FOR X=1TO7:PRINT "DATA":
1010 FOR Y=1TO9:PRINT V((X-1)*9+Y)"||,";
1020 NEXT Y:PRINT "|| ":NEXT X:PRINT
1030 PRINT "DA UNA TECLA POR FAVOR."
1040 GOSUB 940
1050 GOTO 310
1060 :
1070 REM *** CAMBIO DE COLORES ***
1080 C=PEEK (AB+L) AND 15
1090 C=C+1:IF C>15 THEN C=0
1100 POKE AB+L,C
1110 PRINT "XXXX":IF L=33 THEN 1150
1120 PRINT:IF L=37 THEN 1150
1130 PRINT:IF L=39 THEN 1150
1140 PRINT
1150 PRINT TAB(5);C$(C);LEFT$( " ",3-LEN(C$(C)))
1160 RETURN
1170 :
1180 REM *** INSTRUCCIONES PELIGROSAS ***
1190 GOSUB 920
1200 IF N$<>"S" THEN 750
1210 :
1220 POKE 198,0
1230 IF A$="A" THEN 190
1240 IF A$="T" THEN 1550
1250 REM *** SALVAGUARDIA / RECUPERACION K7 ***
1260 PRINT "J":POKE AB+21,0
1270 INPUT "NOMBRE DEL SPRITE ",N$:PRINT
1280 IF A$="C" THEN 1360
1290 :
1300 REM *** SALVAGUARDIA ***
1310 OPEN 1,1,1,N$
1320 FOR X=1TO63:PRINT#1,V(X):NEXT
1330 CLOSE 1
1340 GOTO 260
1350 :
1360 REM *** RECUPERACION ***
1370 OPEN 1,1,0,N$
1380 FOR X=1TO63:INPUT#1,V(X):NEXT
1390 CLOSE 1
1400 :
1410 REM *** CALCULOS TRAS CARGA ***
1420 PRINT "UN MOMENTO POR FAVOR"
1430 Y=0
1440 FOR L=1TO21:FOR X=0TO2
1450 Y=Y+1
1460 FOR C=2TO8STEP 2
1470 Q=X*8+C:P=2*(8-C)
1480 E=V(Y) AND (P*3)
1490 D%(L,Q)=58:D%(L,Q-1)=58
1500 IF E>0 THEN D%(L,Q)=E/P+48:D%(L,Q-1)=D%(L,Q)
1510 NEXT C,X,L
1520 E=1039
1530 GOTO 260
1540 :
1550 REM *** TERMINA ***
1560 POKE AB+21,0:POKE AB+28,0
1570 PRINT "XXXX"
1580 END

```



READY.

© AP 83

Diagrama 1

128	64	32	16	8	4	2	1
1		0		3		2	
01		00		11		10	
0	1	0	0	1	1	1	0
	64			+ 8	+ 4	+ 2	

- ← Valores decimales
- ← Diseño a obtener
- ← Número decimal del color deseado
- ← Valor binario del color
- ← Contenido binario de la dirección
- ← Valor decimal definitivo (=78)

Para obtener este dibujo, que representa uno de los 63 octetos necesarios, el valor de la DATA sord 78

Diagrama 2

128	64	32	16	8	4	2	1
1	0	0	1	0	0	0	1
7	6	5	4	3	2	1	0
128			+ 16				+ 1

- ← Valores decimales
- ← Contenido binario de la dirección 53276
- ← Número de fantasmas
- ← Valor decimal correspondiente

Las instrucciones necesarias para validar el modo multicolor de los fantasmas nº 7, 4 y 0 es:

POKE 53276, 145

Cierto, los fantasmas se vuelven groseros...

Por otra parte, los dos colores suplementarios son los mismos para todos los fantasmas; únicamente el color de base podrá variar de unos a otros, lo que significa que todos los fantasmas tendrán al menos dos colores comunes.

Con un mínimo de imaginación los efectos posibles son extraordinarios a pesar de estas dos restricciones.

El diagrama número 1 os servirá de ayuda en el caso de realizar la codificación a mano. Sin embargo, la utilización del programa generador os aportará la rapidez y comodidad necesarias para este pesado trabajo.

Después de la dirección 53248 (dirección de referencia a partir de la cual se calculan el resto de las direcciones indispensables), ocuparemos la dirección 53276 para definir qué fantasmas serán multicolores.

El contenido de esta dirección es calculado como ya sabemos.

Ver el diagrama número 2.

A continuación, las direcciones 53285 y 53286 contienen los valores numéricos de los colores suplementarios añadidos a los espectros y elegidos de entre los 16 disponibles (valores 0 a 15).

Bien entendido las direcciones utilizadas por los fantasmas monocromáticos siguen siendo válidas.

La utilización del programa generador nos facilitará la tarea.

El editor de fantasmas multicolores propuesto proporciona las facilidades necesarias para la aparición de espectros de buen aspecto: después de su introducción en memoria, y a continuación del encantador RUN habitual, aparecerá a la derecha de la pantalla una matriz de 21 x 24 provista de un cursor móvil en la parte superior. El desplazamiento de este cursor se hará de la forma habitual (teclas de desplazamiento de cursor).

Para encender un punto que forme parte de vuestro fantasma, pulsad los números 1, 2 ó 3. Los colores correspondientes a cada uno de estos números están indicados a la izquierda de la pantalla. Para modificarlos podeis utilizar las teclas de función de la forma siguiente:

- * F1 color de fondo,
- * F3 color suplementario número 1,
- * F5 color propio del fantasma,
- * F7 color suplementario número 2.

Los colores aparecen y desaparecen a voluntad

Cada presión sobre una de estas teclas, visualizará el color correspondiente a la izquierda de la pantalla. La barra espaciadora

Tabla

Dir.	+	Contenido
53248	0	Coordenada x del fantasma 0
53249	1	Coordenada y del fantasma 0
53250	2	Coordenada x del fantasma 1
53251	3	Coordenada y del fantasma 1
53252	4	Coordenada x del fantasma 1
53253	5	Coordenada y del fantasma 2
53254	6	Coordenada x del fantasma 3
53255	7	Coordenada y del fantasma 3
53256	8	Coordenada x del fantasma 4
53257	9	Coordenada y del fantasma 4
53258	10	Coordenada x del fantasma 5
53259	11	Coordenada y del fantasma 5
53260	12	Coordenada x del fantasma 6
53261	13	Coordenada y del fantasma 6
53262	14	Coordenada x del fantasma 7
53263	15	Coordenada y del fantasma 7
53264	16	Coordenada x de mayor peso
53269	21	Selección de los fantasmas
53271	23	Ampliación vertical
53276	28	Modo multicolor
53277	29	Ampliación horizontal
53280	32	Color del borde de la pantalla
53281	33	Color de fondo de la pantalla
53285	37	Color suplementario 1
53286	38	Color suplementario 2
53287	39	Color del fantasma 0
53288	40	Color del fantasma 1
53289	41	Color del fantasma 2
53290	42	Color del fantasma 3
53291	43	Color del fantasma 4
53292	44	Color del fantasma 5
53293	45	Color del fantasma 6
53294	46	Color del fantasma 7

os permitirá borrar un punto previamente encendido. Pulsando la tecla «=» podréis ver el aspecto actual del fantasma en la parte baja de la pantalla. Las teclas H y V os permitirán modificar su tamaño. También podréis visualizar los sesenta y tres elementos de la DATA que compone vuestra criatura pulsando la tecla D.

En este momento, bien tomando nota, o pulsando STOP y después NEW (con la precaución de no borrar la pantalla) podréis colocar un número delante de cada línea DATA creada y hacer un programa con esta base de partida.

Para crear varios fantasmas, podréis irlos salvando y cargando en casete (teclas S y C), lo que os permitirá conservar el trazado de todas vuestras criaturas infernales sin que ninguna se pierda en el más allá...□

Juan Pedro Lalevé

Espíritu ¿estás aquí? o los fantasmas del Commodore 64 (3ª Parte)

Después de los anteriores artículos aparecidos en el OP, ya no tendréis problemas para llevar a cabo la creación de espíritus (amigos) surgidos de las profundidades tenebrosas del Commodore 64. Su comportamiento al desplazarse por la pantalla y encontrar alguno de sus congéneres, es el tema del presente artículo.

Antes de abordar la parte técnica del problema, preparemos el terreno introduciendo en memoria el programa de demostración, que está destinado a generar dos fantasmas reducidos a su más simple expresión.

Bien entendido, las indicaciones dadas en forma de REM xxx.— pueden ser eliminadas del teclado.

Después del RUN fatídico, aparecen dos fantasmas ligeramente camuflados. Entonces puede comprobar que el fantasma de color negro (que es el fantasma nº 0) aparece delante del fantasma nº 2, que es blanco. Si el fantasma nº 1 ha sido generado y situado en su proximidad, aparece en Sandwich entre los dos anteriores.

De este fenómeno se deriva un teorema importante: «los fantasmas que tengan número más bajo aparecen siempre por encima de los fantasmas de número más elevado cuando se les superpone».

Esta regla fundamental le permitirá elegir juiciosamente los números de sus fantasmas, en función del efecto que desea obtener.

En un principio nuestros fantasmas van a encontrarse. Después de haber lanzado el programa por RUN, y sin otra forma de proceso, teclee en modo directo: PRINT PEEK (53278).

El ordenador expone entonces el valor: 5.

A continuación, separe los dos fantasmas tecleando: POKE

53248,200, y vuelva a pedir: PRINT PEEK (53278).

El valor expuesto se convierte en 0. No obstante, para obtener este valor, es posible que tenga que teclear una segunda vez esta misma línea.

Pero antes, recurra al diagrama 1, que le explicará, mejor que un largo discurso, cómo varía el contenido de la dirección 53278.

Los bits que corresponden a los números de los fantasmas en colisión son puestos en 1 y conservan este valor, incluso si la colisión está terminada. Los bits vuelven a tomar su valor normal de 0 en el momento en que se efectúa una lectura en la dirección 53278, para volver a 1 si la colisión existe aún. Esta dirección guarda el recuerdo de los encuentros inter-fantasmas, hasta que sean puestos en acción. Gracias a la misma, sabrá si ha habido encuentro entre ellos y reconocerá a los protagonistas...

¡A mi derecha, el fantasma

Stanley, y a mi izquierda, el fantasma Livingstone!

Para experimentar las colisiones entre fantasmas y texto, borre la pantalla, apoyando simultáneamente las teclas STOP y RESTORE; después relance el programa por RUN.

Haga desaparecer a continuación el fantasma nº 0 tecleando en modo directo: POKE 53269,4.

Finalmente, teniendo cuidado de no escribir sobre una línea de la pantalla ocupada en parte por el fantasma visible, teclee: PRINT PEEK (53279).

El ordenador expone el valor 0.

Ahora, situándose sobre una de las líneas ocupadas por el fantasma, teclee de nuevo la misma instrucción. El valor expuesto ha pasado a 4.

Consulte el diagrama nº 2, que detalla la situación, y observe la similitud con el diagrama precedente.

En el transcurso de este experimento, ha podido notar que el texto inscrito en la pantalla a nivel de un fantasma era enmascarado por este último. Sin embargo, es posible invertir esta situación y superponer el texto sobre el fantasma, merced al contenido de la dirección 53275.

Antes de consultar el diagrama 4, que contiene la clave del misterio, teclee otra vez en modo direc-

PROGRAMA

```
10 AB=53248
15 POKE AB+21,5:REM VALIDACION DE 0 Y 2
20 POKE 2040,14:REM BLOQUE DE DIBUJO DEL FANTASMA 0
25 POKE 2042,14:REM BLOQUE DE DIBUJO DEL FANTASMA 2
30 FOR Q=0 TO 62:POKE 696+Q,255:NEXT:REM RELLENO DEL BLOQUE 14
35 POKE AB+39,0:REM FANTASMA 0 NEGRO
40 POKE AB+41,1:REM FANTASMA 2 BLANCO
45 POKE AB+23,5:POKE AB+29,5:REM EXPANSIONES V Y H
50 POKE AB+1,200:POKE AB+3,180:REM POSICION VERTICAL DE LOS FANTASMAS
55 POKE AB,100:POKE AB+4,70:REM POSICION HORIZONTAL DE LOS FANTASMAS
60 END
```

READY.

DIAGRAMA 1:

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1
7	6	5	4	3	2	1	0
					4	+	1

Los fantasmas 0 y 2 han colisionado

DIAGRAMA 2:

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0
7	6	5	4	3	2	1	0
					4		

El fantasma 2 ha colisionado con un elemento de la pantalla

DIAGRAMA 3:

128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	1
7	6	5	4	3	2	1	0
	64 + 32						+1

Los fantasmas 6, 5 y 0 pasan por detrás del texto de la pantalla

DIAGRAMA 4:

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	1
7	6	5	4	3	2	1	0
							1

El fantasma 1 está a la derecha de la pantalla

- Valores decimales
- Contenido binario de la dir. 53278
- Número de fantasmas
- Val. decimal correspondiente (=5)

- Valores decimales
- Contenido binario de la dir. 53279
- Número de fantasmas
- Val. decimal correspondiente (=4)

- Valores decimales
- Contenido binario de la dir. 53275
- Número de fantasmas
- Val. decimal correspondiente (=97)

- Valores decimales
- Contenido binario de la dir. 53264
- Número de fantasmas
- Val. decimal correspondiente (=1)

to: POKE 53275,4, el texto reaparecerá por encima del fantasma nº 2 (diagrama nº 3).

Los desplazamientos de los fantasmas han sido objeto de largos estudios en Gran Bretaña. En el caso del Commodore 64, el valor posible del desplazamiento horizontal de cualquier fantasma puede variar de 0 a 255, valor que está inscrito en la dirección de coordenada X del fantasma en cuestión.

La pantalla del Commodore 64 ofrece 320 posiciones posibles horizontalmente. Desgraciadamente, toda tentativa de la clase: POKE 53248,320 se salda evidentemente con un mensaje de error... ya que existe un bit de más, que la dirección no puede contener.

Pero, tranquilícese ¡nuestros fantasmas no están condenados a frecuentar solamente la parte izquierda de la pantalla!

Ectoplasmas que se pasean, que se divierten desapareciendo

Basta con situar el bit suplementario en la dirección 53264. Intentemos la experiencia: pulse STOP/RESTORE para limpiar la pantalla y relance el programa en memoria.

Teclee a continuación: POKE 53248,25., el fantasma nº 0 se desplaza hacia la parte extrema izquierda de la pantalla.

Finalmente, desde el momento

en que teclee: POKE 53264,1, este mismo fantasma podrá abordar sin dificultades la zona prohibida (diagrama 4).

Tengo cuidado, si modifica los valores propuestos anteriormente es posible que su fantasma salga de la pantalla por la derecha y, por consiguiente, se hace invisible. En este caso, disminuye el valor dado en 53248 (coordenada X del fantasma).

Para terminar, completemos el programa en memoria:

READY.

```
60 FOR Q=0TO255
65 POKE AB+4,Q
70 NEXT
75 POKE AB+16,4
80 FOR Q=0TO70
85 POKE AB+4,Q
90 NEXT
100 POKE AB+16,0
105 GOTO 60
```

READY.

El fantasma nº 2 circulará entonces de izquierda a derecha en la pantalla.

Durante el funcionamiento del programa, observe la intermitencia del fantasma durante su paso por la línea-frontera, y sus aplicaciones fugitivas hacia lugares no deseados. Estos defectos son debidos a la imperfección del BASIC, demasiado lento para este empleo. En próximo artículo veremos cómo paliar este inconveniente utilizando el lenguaje máquina, con

DIR.	+	CONTENIDO
53248	0	COORDENADA X DEL FANTASMA 0
53249	1	COORDENADA Y DEL FANTASMA 0
53250	2	- X - 1
53251	3	- Y - 1
53252	4	- X - 2
53253	5	- Y - 2
53254	6	- X - 3
53255	7	- Y - 3
53256	8	- X - 4
53257	9	- Y - 4
53258	10	- X - 5
53259	11	- Y - 5
53260	12	- X - 6
53261	13	- Y - 6
53262	14	- X - 7
53263	15	- Y - 7
53264	16	PESO FUERTE DE LAS COORDENADAS X
53269	21	VALIDACION DE LOS FANTASMAS
53271	23	EXPANSION VERTICAL DE LOS FANT.
53275	27	PRIORIDAD TEXTO/FANTASMAS
53276	28	SELECCION DE FANT. MULTICOLORES
53277	29	EXPANSION HORIZONTAL DE LOS FANT
53278	30	COLISION ENTRE FANTASMAS
53279	31	COLISION TEXTO/FANTASMAS
53280	32	COLOR DEL BORDE DE LA PANTALLA
53281	33	COLOR DEL FONDO DE LA PANTALLA
53285	37	COLOR 0 EN MODOS MULTICOLOR
53286	38	COLOR 1 EN MODOS MULTICOLOR
53287	39	COLOR DEL FANTASMA NUMERO 0
53288	40	- 1
53289	41	- 2
53290	42	- 3
53291	43	- 4
53292	44	- 5
53293	45	- 6
53294	46	- 7

un programa de juego muy completo que pone en práctica elementos estudiados en el curso de los artículos precedentes.

Jean Pierre Lalevée

Espíritu, ¿estás ahí? Los fantasmas del Commodore 64

Para concluir, quizás provisionalmente, nuestra revista de los fantasmas errantes en el Commodore 64, les proponemos un programa completo que emplea la mayor parte de los elementos tratados en los artículos anteriores. Por él pueden conocer algunos trucos complementarios indispensables.

Aunque no nos haya seguido por las abruptas pendientes de la técnica, la memorización de nuestro programa le proyectará a la naturaleza, donde no temerá más que una cosa: que le parta un rayo.

Bajo un cielo amenazador, cubierto de negros cúmulos que precipitan pedriscos destructores, debe proteger magníficas verduras. Para ello, sólo dispone de un paraguas, cuya superficie es demasiado pequeña y le obliga a correr de acá para allá entre los vegetales para abrigar durante cierto tiempo a estas plantas indefensas.

Para reforzar todo ello, relámpagos acompañados de terroríficos truenos desgarran el cielo con intención de fulminarle. ¡Desgraciado si no huye a tiempo!

Desencadenado el torbellino de los elementos, no podrá evitar la brutal muerte de algunos. Por fortuna, rebrotan rápidamente. Pero, ¿podrá realizar una buena cosecha?

Un valiente hortelano defiende las lechugas

Para que el juego sea lo más real posible, hemos adoptado soluciones

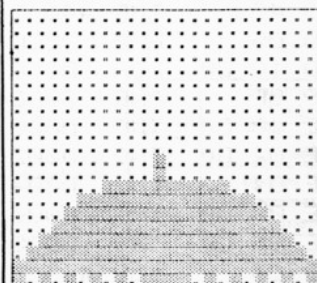
técnicas relativamente complejas: el valiente hortelano defensor de las lechugas, está formado por un conjunto de varios fantasmas multicolores elementales yuxtapuestos, cuyo desplazamiento se asegura mediante un sub-programa escrito en lenguaje máquina.

Es evidente, que el empleo del Basic haría mucho más lento el control del conjunto de desplazamientos simultáneos necesarios.

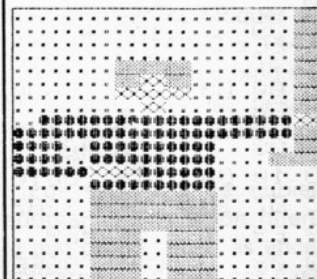
• **Los fantasmas.**— El esquema n.º 1 muestra una representación de cada fantasma elemental, acompañado de su número y del número del bloque de memoria en el que está memorizado el dibujo.

• **El problema de los bloques.**— Para memorizar el dibujo de cada fantasma debemos emplear ocho bloques de memoria capaces de contener 63 octetos cada uno. Sin embargo, se plantea un problema: la organización de la memoria del Commodore 64 permite disponer de solo cuatro bloques sin manipulación particular (el bloque 11 que está de la dirección 704 a 766 y los bloques 13, 14 y 15 que ocupan la totalidad del tampón cassette y van de la 828 a 1022). Nos es preciso, por tanto, encontrar otros emplazamientos. La dificultad

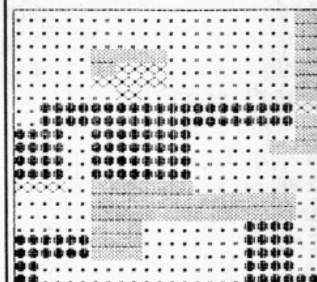
Esquema n.º 1



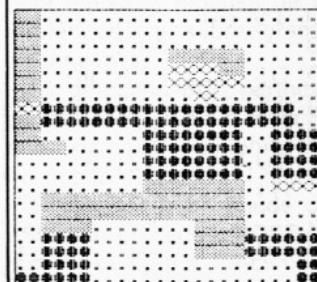
Fantasma
n.º 0,
bloque
220,
expansión
X,
monocolor.



Fantasma
n.º 1,
bloque
221,
expansión
Y,
multicolor.



Fantasma
n.º 2,
bloque
222,
expansión
Y,
multicolor.



Fantasma
n.º 3,
bloque
223,
expansión
Y,
multicolor.

SUB PROGRAMA

[illegible]


```

475 Z=PEEK(E):Z1=PEEK(E+1)
480 IF(Z=950RZ=105) OR (Z1=950RZ1=105) THEN 490
485 POKEE,95:POKEE+1,105:POKEE+J,5:SC=SC+20
490 SC=FNX(SC):PRINTSC$:SC=" "
495 NEXTR
497 :
500 REM *****FIN DE LA PARTIDA *****
505 IFSC>SMTHENSM=SC
510 PRINT"XXXXXXXXXX","MEJOR TANTEO: "SM" "
515 POKE190,0:PRINT"OTRA TORMENTA (S/N)?":POKE204,0
520 GETR$:IFR$="S" THEN SC=0:POKE204,1:GOTO190
525 IF R$<>"N" THEN 520
530 PRINT" ":POKE 54296,0
540 POKE 808,237
545 END
599 :
600 REM RELAMPAGO -----
605 GOSUB800:IFRND(0)<.5 THEN 660
610 Y=50:X=100+155*RND(0):POKES+10,X:POKES+11,Y
615 Z=2*(RND(0)<.5)+1
620 FORW=0TO10:POKES2,PEEK(S2)+32:Y=Y+15
625 X=X+Z*(3+3*RND(0)):IFX<0ORX>255 THEN Z=-Z:X=X+Z*6:GOTO635
630 POKES+10,X:POKES+11,Y
635 POKES2,PEEK(S2)AND223
645 SYSDX:POKES1,PEEK(S1)+32
650 IFW>5 THEN K=PEEK(S4):IF(KAND1) THEN 700
651 NEXT
655 POKES1,PEEK(S1)AND223
660 POKES4,PEEK(S4)AND31:POKES5,0
665 RETURN
670 :
700 REM EL JARDIN ES FULMINADO -----
705 SC=SC-100:PRINTSC$:FNX(SC):" "
710 POKES1,211:POKES4,0
715 FORQ=0TO1:POKES+39,Q:POKES+40,Q:POKES+43,Q:POKES+37,Q:POKES+38,Q
720 POKE54277,0:POKE54278,240
725 POKE54273,51:POKE54272,97:POKE54276,17:FORP=1TO50:NEXTP
730 POKES+39,4:POKES+40,6:POKES+43,6:POKES+37,8:POKES+38,1
735 POKE54273,25:POKE54272,177:POKE54276,17:FORP=1TO50:NEXTP
740 NEXTQ:GOTO960
750 :
800 REM TRUENO -----
805 POKE54277,190:POKE54278,240:POKE54276,129
810 FORQ=20TO59STEP1,5:POKE54296,2+11*RND(0)
815 POKE54273,61:POKE54272,126
820 SYSDX:NEXTQ
825 FORQ=15TO55STEP-2
830 POKE54296,0:POKE54277,15:POKE54278,136
835 POKE54273,40:POKE54272,200:POKE54276,129
840 SYSDX:NEXTQ
845 POKE54296,15:GOTO960
850 :
900 REM RUIDO 1 -----
905 POKE54277,0:POKE54278,240
910 POKE54273,34:POKE54272,75:POKE54276,17:GOTO950
915 :
930 REM RUIDO 2 -----
935 POKE54277,0:POKE54278,240
940 POKE54273,64:POKE54272,75:POKE54276,17
950 REM INTERRUPCION DEL SONIDO ----
955 FORQ=1TO25:NEXT
960 POKE54276,0:POKE54277,0:POKE54278,0
965 RETURN
999 :
1000 REM PRESENTACION DEL JUEGO-----
1005 POKE 0,6:POKE F,3
1010 PRINT"CHR$(14):PRINT"
1015 PRINT"EL CIELO SE CUBRE..."
1020 PRINT"AYUDA AL PEQUEÑO HORTELANO A PROTEGER SUS LECHUGAS."
1025 PRINT"PULSA "
1030 PRINTTAB(8)"PARA IR A IZQUIERDA"
1035 PRINTTAB(8)"DEL PARA IR A DERECHA"
1040 PRINTTAB(8)"XXXXXXXXXXXX"
1045 RETURN
1050 :
1100 REM BLOQUES GRAFICOS DE 6 FANTASMAS
1105 REM PARAGUAS=FANTASMA 0 #####
1110 AX=14080:FORQ=0TO32:POKEAX+Q,0:NEXT
1115 FORQ=33TO62:READFX:POKEAX+Q,FX:NEXT
1120 POKE2040,220:GOTO1200
1125 DATA 0,16,0,0,16,0,1,255,128,7,255,224,15,255,240,31,255
1130 DATA 240,63,255,252,127,255,254,255,255,255,170,189,85
1135 :
1200 REM HORTELANO INMOVIL = FANTASMA 1 #####

```



nes 55/56 y 51/52). De tal modo, podemos liberar al final del Basic un espacio suficiente.

Fin de las cadenas

Basic (1)	Sistema y bloques 11, 13, 14 y 15
Basic (2)	Programa, tablas y cadenas
Libre	Bloques 220 a 225 Subprogramas de desplazamientos

Por razones prácticas, hemos decidido emplear los bloques 220 a 225 (direcciones 14080 a 14464) y conservar el uso de los bloques 13 y 14. El subprograma de desplazamiento de los fantasmas comienza en la dirección 14500.

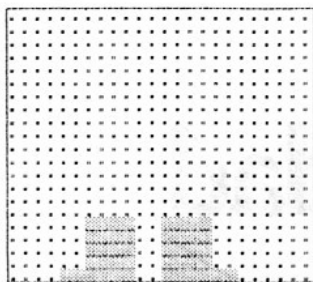
Por supuesto, se hubiera podido adoptar una solución diferente, pero la que proponemos es eficaz y puede aplicarse en numerosos casos.

• **El problema de los desplazamientos.**— Los aficionados al lenguaje máquina encontrarán un programa sencillo, que cumple perfectamente su cometido. Pueden hacerse algunas modificaciones pequeñas para emplear, por ejemplo, una manecilla de juego en vez de las teclas.

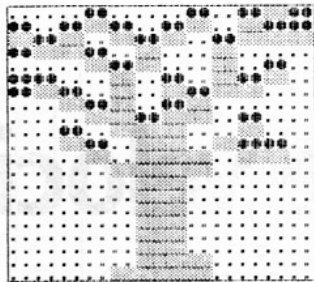
• **Análisis del programa.**— Línea 25: la modificación de los apuntadores del final de Basic y de final de cadenas libera un espacio, disponible para nuestros bloques de fantasmas numéricos y el subprograma de desplazamientos.

La dirección 808 permite neutralizar la pulsación de la tecla STOP para que una falsa maniobra del jugador no pueda perturbar el desarrollo normal del juego. Se puede suprimir el REM que precede a POKE en esta dirección cuando el funcionamiento del programa esté verificado.

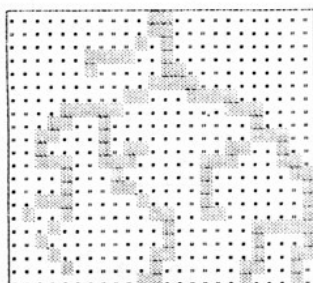
Esquema n.º 2



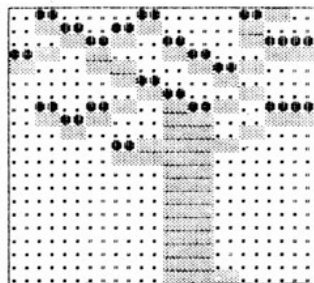
Fantasma n.º 4, bloque 224, expansión Y, monocolor.



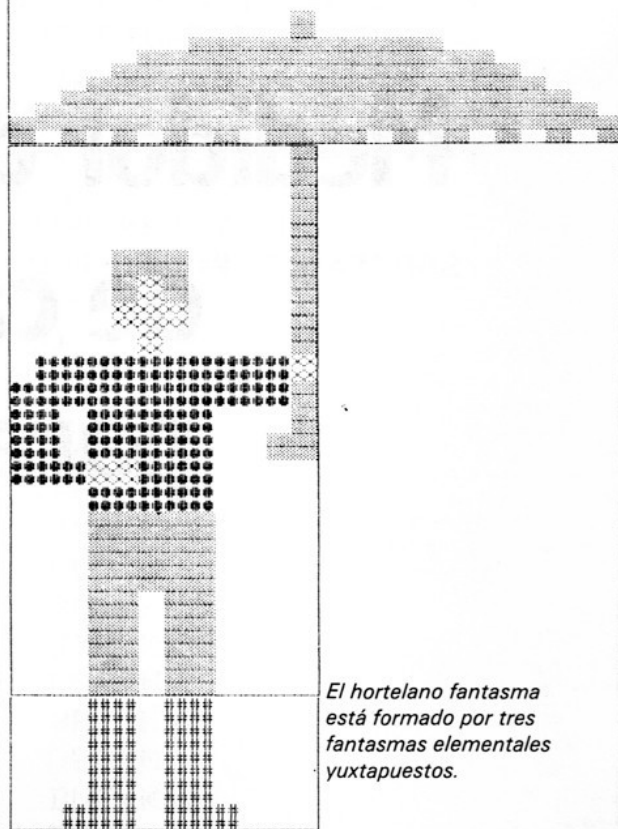
Fantasma n.º 5, bloque 225, expansión X permanente, monocolor.



Fantasma n.º 6, bloque 13, multicolor.



Fantasma n.º 7, bloque 14 multicolor.



El hortelano fantasma está formado por tres fantasmas elementales yuxtapuestos.

de un enmascaramiento lógico en la línea 380. Si el granizo alcanza el suelo sin colisionar con el paraguas, eventualmente puede aplastar una verdura y su dibujo se modifica en la pantalla. Las líneas 395 a 405 permiten amontonar la nieve si llega el caso. El choque del granizo con el paraguas o con el suelo se acompaña con un sonido musical asegurado por dos subprogramas, llamados según el caso. Efectuada la actualización del marcador, el programa decide aleatoriamente si debe resonar el estruendo siniestro del trueno (línea 455) tras lo que examina el eventual rebrote de una verdura.

Líneas 500 a 599: final de partida. Se memoriza y presenta la mejor puntuación y se propone una nueva partida.

Líneas 600 a 655: resuena un trueno. Puede estar seguido de un relámpago, cuyo emplazamiento en la pantalla se determina aleatoriamente y sigue un trayecto sinuoso. Su descenso rápido hacia el suelo está acompañado por una sucesión de expansiones y contracciones del fantasma, haciendo su aspecto más realista. La llamada al subprograma de desplazamiento (línea 645) permite huir al hortelano. La línea 650 detecta la colisión posible entre uno de los fantas-

mas (el hortelano o el relámpago). Si el hortelano no ha sido fulminado, el programa prosigue normalmente por una simple vuelta al programa principal.

Epitafio por un pobre hortelano fulminado

Líneas 700 a 740: si el hortelano ha sido alcanzado, el tanteo disminuye cien puntos; los fantasmas parpadean varias veces cambiando de color y suena una música.



Líneas 800 a 965: contienen los tres subprogramas musicales empleados frecuentemente.

Líneas 1100 a 1685: este subprograma se divide en ocho partes iguales para cada uno de los fantasmas.

Los DATA contienen el dibujo digitalizado, que se ubica en un adecuado bloque de memoria.

Líneas 1700 a 1810: características de los fantasmas (tamaño, colores, situación al principio de la partida, etc.).

Líneas 2000 a 2045: el subprograma de desplazamientos tiene 135 octetos colocados a partir de la dirección 14500, o sea 38A4 en hexadecimal. Vigile que no se produzca ninguna equivocación.

Antes de lanzar el programa, efectúe una salvaguarda en banda o disquette.

Un error cometido en la transcripción, en especial en el subprograma de desplazamientos, le podría ocasionar la pérdida de control de su máquina.

En este caso, solamente un RESET seguido de una nueva impresión de este largo programa le permitiría disponer de él de nuevo.

Juan Pedro Lalevée

Medidor de velocidad de cassette para Atom

Si alguna vez estuviste tratando de introducir un programa desde una cassette prestada sin conseguirlo, estas notas te interesan.

Uno de los problemas con que se encuentra el iniciado en los microordenadores caseros es que no dispone de minifloppy para almacenar programas y ha de asistirse del cassette en el intercambio de cintas con otros usuarios.

Normalmente en sistemas de bajo precio, al diferir la velocidad de los grabadores y también al haber diferencias en la altura de la cabeza de grabación, puede llegar a resultar enojoso el intercambio de cintas entre usuarios.

Respecto al segundo problema multitud de grabadores pequeños, Sanyo u otros, llevan un tornillo que permite elevar o bajar la cabeza y por medio de un medidor ó simplemente un auricular podremos obtener un óptimo en cuanto al volumen de la señal.

Respecto al primer problema de velocidad de la cinta es lo que en éstas líneas trataremos de resolver por medio de un programa que nos medirá la frecuencia de la portadora de FSK que en el caso del ACORN ATOM es de 2.400 hz ya sea para transmitir a 300 bd ó 1200 bd que es el máximo de ve-

locidad posible con 2.400 hz de frecuencia base.

Cuando en el ACORN ATOM deseamos salvar algún programa normalmente hacemos SAVE "nombre fichero", después damos RETURN y vemos el mensaje RECORD TAPE, pues bien en este momento y hasta que volvamos a pulsar RETURN el microordenador está enviando al cassette el tono de 2400 hz que podemos estar grabando durante un espacio de tiempo hasta que al dar RETURN de nuevo comienza a enviar los datos a la cinta. Es precisamente esa grabación de 2400 hz la que el siguiente programa va a medir para darnos en la reproducción su valor y si la diferencia obtenida es superior al $\pm 5\%$ hemos de suponer que no vamos a recuperar el fichero.

Para superar este problema hemos de regular la velocidad del grabador/reproductor por medio de un potenciómetro más o menos accesible que suelen llevar, hasta el valor deseado 2.400 hz.

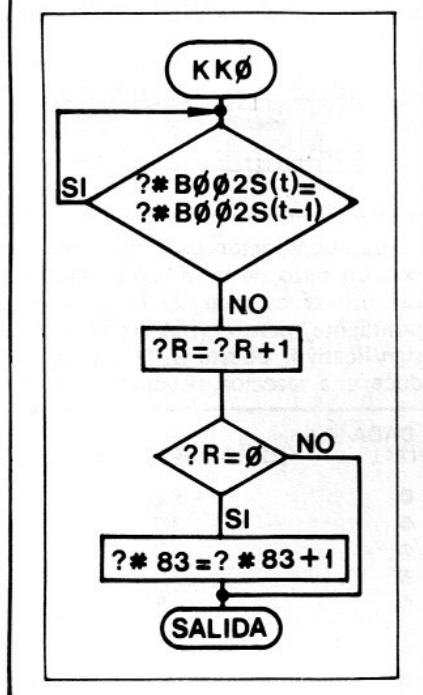
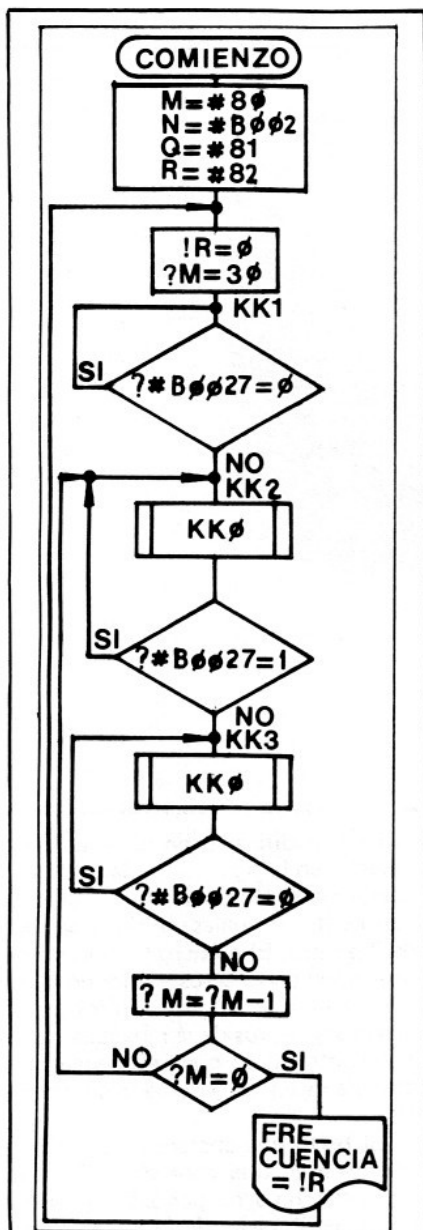
Para su funcionamiento después de la carga no tenemos más que efectuar RUN y de esta forma queda a la espera de que le entre la señal desde la cinta para monitorizar de que frecuencia es.

Para su parada hacemos BRK y después con OLD lo recuperaremos de nuevo.

DESCRIPCION DEL PROGRAMA

Según podemos ver por el esquema de ACORN ATOM y por el manual de BASIC en su página 194, el microordenador dispone de un C.I. 8255 que es un P.P.I.A. (Programable Peripheral Interface Adapter) con el cual gestiona el teclado y la entrada/salida al cassette.

Este circuito integrado dispone de tres puertas A B C y en el bit 5 de este último (llamado ? # B 0 0 2₅ en el esquema de flujo) es por donde recibe la entreada de audio del cassette, similarmente por el bit 7 (? # B 0 0 2₇) la P.P.I.A. recoge una señal de V.D.G. (Video Generator Display C.I. 6847) que es el Flyback (FS). Esta señal da un impulso negativo de 2'5 mseg cada vez que el haz electrónico del monitor de video pasa de la última línea del cuadro a la primera del siguiente cuadro, mientras que el período de esta señal al ser la frecuencia de cuadro para el sistema NTSC de 60 hz o sea tiene un período de 16'6 mseg.



```

>L. 1REM*****
2REM* MEDIDOR VEL.CASSETTE*
3REM* AUTOR F.GUTIERREZ *
4REM* COPYRIGHT EL AUTOR *
5REM* Y EL ORDENADOR PER *
6REM* SONAL *
7REM*****
8DIMKK4
9F.J=0 TO4;KKJ=T.;N.
10M=#80;Q=#81;R=#82;N=#B002
11P.#21
20F.J=0TO1
30DIMP-1
40C
50:KK1 LDA N
60 BPL KK1
70:KK2 JSR KK0
80 LDA N
90 BMI KK2
100:KK3 JSR KK0
110 LDA N
120 BPL KK3
130 DEC M
140 BNE KK2
150 RTS
200\\RUTINA CUENTA FLANCOS
210:KK0 LDA N
220 AND @#20
230 CMP Q
240 BEQ KK0
250 STA Q
260 INC R
270 BNE KK4
280 INC R+1
290:KK4 RTS
300]
310N.;P.#6;DO !R=0
320?M=30;LINKKK1
330IF !R>2412 OR !R<2382 G.350
340P."FRECUENCIA="!R" Hz";U.0
350P." frecuencia="
360P.!R" Hz";U.0;E.

```

Seguindo el esquema de flujo de la figura 1 vemos que $?*B005(t) = ?*B002_5(t-1)$ significa que el bit de 5 de la dirección $?*B002$ es igual en el ciclo actual t que en el anterior $t-1$.

La rutina $KK0$ cuenta flancos de la señal de audio y así vemos como el programa espera el primer flyback $\overline{FS}$ para comenzar la cuenta sobre $?R$ y $?R+1$. Hace el recuento con $\overline{FS} = 0$ y después con $\overline{FS} = 1$ repitiendo el proceso 30 veces dando después el re-

sultado en el monitor de video (un período tiene 2 flancos).

Como el programa está dirigido a usuarios del ATOM el direccionamiento indirecto está expresado por $?$ para byte y $!$ para palabra (4 bytes) en la FIG.1. siendo $\#$ el signo que indica hexadecimal y $KK0, \dots, KK4$ son etiquetas.

Confío que esta rutina os evite pérdidas de tiempo en el futuro.

F. Gutierrez.

Programa para alta resolución en el ZX81

NOTA PARA EL ARTICULO DE ALTA RESOLUCION ZX81

El objeto de este artículo es especialmente didáctico, aunque esperamos que como programa sea de interés para muchos usuarios del ZX81.

Su confección se realizó en Diciembre del año pasado, posteriormente se han mejorado algunos aspectos. Un programa de este tipo pero más acabado, incluyendo representaciones tridimensionales. Se acaba de comercializar recientemente en Inglaterra.

La instrucción PLOT permite realizar gráficas con el ZX81 en la pantalla del televisor, con una resolución de 64X44 puntos "cuadrados", cada uno de los cuales tiene el grosor de cuatro líneas de barrido. El programa que se propone en el presente artículo permite reducir el grosor de los puntos al de una línea de barrido, incrementando la resolución a 192X178 puntos.

El proceso de visualización (1)

La imagen en un televisor se forma mediante un punto luminoso de intensidad variable que recorre rápidamente la pantalla trazando líneas horizontales de izquierda a derecha y de arriba a abajo. Unos impulsos especiales de sincronismo se encargan de hacer regresar el punto luminoso a la izquierda después del trazado de cada línea, y de llevarlo a la parte superior cuando se ha completado un cuadro.

El proceso de visualización en el ZX81 consta de tres etapas. En la primera se generan las líneas en blanco del margen superior de la pantalla. En la segunda se trazan las líneas correspondientes a la zona donde aparece la imagen del texto. En la tercera se generan las líneas blancas del margen inferior. Estas operaciones son realizadas por las rutinas de la ROM encargadas de esta tarea. Cuando el ordenador trabaja en modo SLOW, las rutinas de visuali-

(1) Puede encontrarse una detallada explicación del proceso de visualización del ZX81 en el libro de Mike Lord: "The Explores Guide to the ZX81" (Timedata).

zación entran en acción mediante interrupciones periódicas.

Cada carácter se visualiza en pantalla disponiendo pautas de puntos en una cuadrícula de 8X8, de modo que

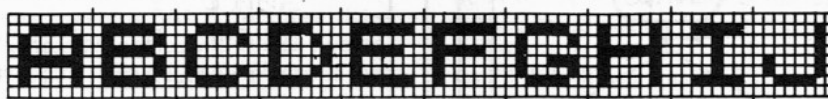


Fig. 1

cada fila de caracteres exige el trazado de 8 líneas (fig. 1).

Las pautas de puntos que componen cada carácter se encuentran en el generador de caracteres al final de la ROM, entre las posiciones 1E00 h y

1FFFh, codificadas en la forma que se muestra en la figura 2. Cada carácter se divide en 8 rebanadas horizontales, cada una de las cuales contiene una pauta de 8 puntos. El número binario obtenido disponiendo ceros y unos en la misma forma que se disponen los puntos blancos y negros de la rebanada correspondiente, es lo que se encuentra almacenado en el generador de caracteres.

El texto que aparece en pantalla se almacena en una zona de la RAM llamada "archivo de pantalla" ("display file"), codificado de la forma que se indica en el apéndice A del manual del Sinclair. Cada fila de caracteres acaba con un símbolo de NEWLINE (código 76h).

La visualización del texto comienza con un salto de la CPU a la dirección del primer carácter de la fila correspondiente, pero con el bit 15 (el más significativo) puesto a "1". Esto produce una reacción peculiar en el chip

CARACTER	CODIGO BINARIO DE CADA REBANADA HORIZONTAL	CODIGO HEXADECIMAL
	0 0 0 0 0 0 0 0	00
	0 0 1 1 1 1 0 0	3C
	0 1 0 0 0 0 1 0	42
	0 1 0 0 0 0 1 0	42
	0 1 1 1 1 1 1 0	7E
	0 1 0 0 0 0 1 0	42
	0 1 0 0 0 0 1 0	42
	0 0 0 0 0 0 0 0	00

Fig. 2

PROGRAMA PRINCIPAL			
Direcciones	Códigos de operación	Mnemónicos	Comentarios
4082	3E1E	START	LD A,1E
	ED47		LD I,A
	0EFE		LD C, FE
	067F		LD B,7F
	ED78		IN A, (C)
	CB47		BIT 0,A
	C8		RET Z
	3E08		LD A,08
	ED47		LD I,A
	0659		LDB, 59
4095	10FE	LAZ01	DJNZ, LAZ01
	DD2100E1		LD IX, ARCH*
	110000		LD DE,0000
	21B43C		LD HL, 3CB4
	A7		AND A
	D8		RET C
	D8		RET C
	00		NOP
	7E		LD A, (HL)
	0619		LD B,19
	4C		LD C,H
	CDBC40		CALL GEN
	58		LD E,B
	4D		LD C,L
	CDBC40		CALL GEN
	5A		LD E,D
	4C		LD C,H
	CDBC40		CALL GEN
	060B		LD B,0B
40B8	10FE	LAZO2	DJNZ, LAZO2
	18C6		JR START
40BC	DBFE	GEN	IN A,(FE)
	D3FD		OUT (FD), A
	CDCC40		CALL LIN
	7E		LD A,(HL)
	7E		LD A,(HL)
	0D		DEC C
	C8		RET Z
	00		NOP
	DD19		ADD IX,DE
	18F0		JR GEN
40CC	DDE9	LIN	JP (IX)

LISTADO 1

ROUTINA DE BORRADO			
Direcciones	Códigos de operaciones	Mnemónicos	Comentarios
40D2	210061	BORR	LD HL,ARCH
	0E84		LD C,B4
40D7	0618	LAZO4	LD B,18
40D9	369E	LAZO3	LD(HL), 9E
	23		INC HL
	10FB		DJNZ, LAZO3
	36C9		LD(HL), C9
	23		INC HL
	0D		DEC C
	20F3		JR NZ,LAZO4
40E4	C9		RET

LISTADO 2

de lógica (SLC), que fuerza una instrucción NOP en el bus de datos y almacena el código del carácter a visualizar. Durante la segunda mitad del ciclo de la instrucción NOP (dedicado al refresco de la memoria dinámica), el SLC pone el contenido del registro I (1Eh) en los siete bits altos del bus de direcciones (el bits 0 del registro I se ignora). En los seis bits siguientes se pone el contenido de los seis bits bajos del código del carácter a visualizar, y en los tres más bajos el número (0 a 7) de la línea que se está trazando dentro de la fila de caracteres. El SLC toma nota además de si el bit 7 del código del carácter es 1, en cuyo caso la visualización se realizará en negativo (carácter blanco sobre fondo negro). La dirección configurada en el bus de direcciones apunta ahora al generador de caracteres, justo al byte que contiene la pauta de puntos de la rebanada que debe visualizarse del carácter en curso. Dicha pauta se toma y se suministra en serie al modulador de TV.

A continuación la CPU busca su siguiente instrucción en el siguiente byte del archivo de pantalla, y así se repite el proceso hasta que se alcanza el NEWLINE de final de línea. El código de NEWLINE se interpreta como la instrucción HALT (también de códigos 76h), lo que detiene la CPU en espera de una interrupción que marcará el inicio de la siguiente línea de barrido.

Alta resolución en el ZX81

El programa en CM que se propone (listado 1) se ha tomado de un artículo de Ron Bissell (aunque aquí ha sido retocado en varios aspectos), y funciona de forma parecida a como lo hacen las rutinas de visualización de la ROM. Para evitar que éstas interrumpan el programa de alta resolución, la ejecución deberá realizarse en modo FAST.

Este programa presenta las siguientes diferencias esenciales frente a las rutinas de visualización de la ROM:

1. El registro I no contiene 1Eh, sino 08, a fin de tomar las pautas de puntos de la zona 0800h-09FF, en lugar del generador de caracteres. Esto permite manipular pautas distintas de las que contiene el generador de caracteres. Lo ideal sería poder diseñar las pautas y almacenarlas en la RAM, pero esto no funciona porque la memoria dinámica se deshabilita durante el ciclo de refresco, instante que aprovecha el SLC para buscar la pauta de puntos.

2. El dispositivo que lleva la cuenta de qué línea se está trazando dentro de la fila de caracteres (probablemente un contador interno del SLC) es puesto a cero al principio de cada línea mediante una instrucción IN A, (FE). Esto

```

1 REM 12345678901234567890123
45678901234567890123456789012345
67890123456789012345678901234567
8901234567890
5 REM PROGRAMA CARREGA
10 LET A$="3E1EED470EFE087FED7
8CB47C83E08ED47065910FEDD2100E11
1000021843CA7D8D8007E0619"
20 LET A$=A$+"4CCDBC40584DCDBC
405A4CCDBC40060B10FE18C6DBFED3FD
C0CC407E7E0DC800DD1918F00DE9"
30 LET A$=A$+"000000002100610E
B40618369E2310FB36C9230D20F3C9"
500 LET D=16514
510 FOR N=1 TO LEN A$ STEP 2
520 LET A=16*(CODE A$(N)-28)+CO
DE A$(N+1)-28
530 POKE (D+(N-1)/2),A
540 NEXT N

```

LISTADO 3

```

1 REM Y2 GOSUB ? : RETURN G
OSUB ?ACS ?COS Y GOSUB ? ? ( RET
URN <>5 LPRINT ) SUB**** AND?
?LN RAND??LN RAND ( RETURN /LEN
<= RETURN PEEK CLEAR LN ATN AND
/ LIST <> DIM 5 ? : 0.71 C
LS QTAN 7#4 NEXT TAN 0
5 REM PROGRAMA CARREGA
10 LET A$="3E1EED470EFE087FED7
8CB47C83E08ED47065910FEDD2100E11
1000021843CA7D8D8007E0619"
20 LET A$=A$+"4CCDBC40584DCDBC
405A4CCDBC40060B10FE18C6DBFED3FD
C0CC407E7E0DC800DD1918F00DE9"
30 LET A$=A$+"000000002100610E
B40618369E2310FB36C9230D20F3C9"
500 LET D=16514
510 FOR N=1 TO LEN A$ STEP 2
520 LET A=16*(CODE A$(N)-28)+CO
DE A$(N+1)-28
530 POKE (D+(N-1)/2),A
540 NEXT N
1000 SLOW
1010 FAST
1020 IF USA 16514 THEN REM

```

LISTADO 4

```

1 REM Y2 GOSUB ? : RETURN G
OSUB ?ACS ?COS Y GOSUB ? ? ( RET
URN <>5 LPRINT ) SUB**** AND?
?LN RAND??LN RAND ( RETURN /LEN
<= RETURN PEEK CLEAR LN ATN AND
/ LIST <> DIM 5 ? : 0.71 C
LS QTAN 7#4 NEXT TAN 0
10 REM PROGRAMA "CUR.HR6A"
20 REM (MIGUEL A. LERMA/1983)
30 REM COPYRIGHT EL AUTOR Y
40 REM EL ORDENADOR PERSONAL
250 SLOW
250 PRINT
300 PRINT "PROGRAMA ""CUR.HR6A""
310 PRINT
320 PRINT "TRAZADO DE CURVAS"
330 PRINT "EN ALTA RESOLUCION"
340 PRINT
350 PRINT "1. INICIAR Y BORRAR"
350 PRINT "2. INTRODUCIR CURVA"
370 PRINT "3. REVISUALIZAR"
380 PRINT "4. CENTRADO HORIZONT
AL"
390 PRINT "5. MODIFICAR NUMERO
DE LINEAS"

```

```

450 INPUT XX
460 CLS
500 IF XX=1 THEN GOSUB 8000
510 IF XX=2 THEN GOTO 600
520 IF XX=3 THEN GOTO 8100
530 IF XX=4 THEN GOTO 8500
540 IF XX=5 THEN GOTO 8600
570 RUN
600 PRINT "BORRO TODO? (SI/NO)"
605 INPUT X$
607 PRINT X$
610 IF X$>="0" THEN GOSUB 8000
615 PRINT
620 PRINT "INTRODUZCA LA FUNCIO
N EN FORMA"
630 PRINT "EXPLICITA EN LA LINE
A 3000"
640 PRINT "Y LUEGO PULSE RUN 10
00"
650 PRINT
660 PRINT " PARA INTERRUPIR L
A VISUALIZA-CION, PULSE BREAK"
700 STOP
1000 REM
1500 FAST
1700 LET NL=PEEK 16543-1
2000 FOR X=1 TO 191
3000 LET Y=85+70*SIN (X/30)
3050 LET Y=INT (NL-Y)
3100 IF Y<1 OR Y>=NL THEN GOTO 4
000
3200 LET XX=X-INT (X/8)*8
3250 GOTO 3300+10*XX
3300 LET C=158
3305 GOTO 3400
3310 LET C=11
3315 GOTO 3400
3320 LET C=27
3325 GOTO 3400
3330 LET C=158
3335 GOTO 3400
3340 LET C=42
3345 GOTO 3400
3350 LET C=153
3355 GOTO 3400
3360 LET C=15
3365 GOTO 3400
3370 LET C=46
3400 LET DIR=24832+25*Y+INT (X/8
)
3500 POKE DIR,C
4000 NEXT X
5000 RUN 8100
7000 LET Y=85+70*SIN (X/30)
8000 REM "HIGRES"
8020 IF USA 16594 THEN REM
8050 RETURN
8100 SLOW
8200 FAST
8300 IF USA 16514 THEN REM
8350 SLOW
8400 RUN
8500 REM
8520 PRINT "CENTRADO HORIZONTAL"
8540 PRINT PEEK 16567
8550 INPUT HH
8553 PRINT HH
8560 POKE 16567,HH
8570 POKE 16532,100-HH
8580 RUN
8600 REM
8620 PRINT "NUMERO DE LINEAS"
8630 PRINT PEEK 16543
8650 INPUT HH
8655 PRINT HH
8660 POKE 16598,HH
8670 POKE 16543,HH
8680 POKE 16544,INT ((300-HH)/2)
8685 IF USA 16594 THEN REM
8690 RUN
9000 STOP
9980 SAVE "CUR.HR6A"
9995 RUN

```

LISTADO 5

capaz de albergar la función USR 16594.

Programa "CUR. HR6A"

Este programa hace uso del CM descrito para visualizar gráficas de funciones en alta resolución.

Sólo se emplean pautas de puntos compuestas por un único punto en negro para cada carácter-rebanada. En la tabla 1 se comprueba que no es posible colocar un punto único en las posiciones 1 y 4. Para las restantes posiciones se usan los siguientes códigos:

posición	código hex.	código dec.
1	-	--
2	0B	11
3	1B	27
4	--	--
5	2A	42
6	99	153
7	0F	15
8	2E	46

Cuando un punto debe ir en la posición 1 ó 4 del carácter, el programa asigna el código de blanco (9Eh=158d). Esto puede dar un aspecto algo discontinuo a la gráfica, pero el resultado puede considerarse aceptable.

La ejecución del programa comienza con el siguiente menú:

1. INICIAR Y BORRAR
2. INTRODUCIR CURVA
3. REVISUALIZAR
4. CENTRADO HORIZONTAL
5. MODIFICAR NUMERO DE LINEAS.

Debe comenzarse siempre con la opción 1, que se encarga de crear una zona de archivo en blanco. Puede usarse igualmente esta opción para borrar gráficas previamente introducidas.

La opción 2 pregunta primero si se desean borrar o no las gráficas introducidas, y luego informa cómo hacer para introducir una gráfica nueva: debe colocarse la función a representar en la línea 3000 del programa en la forma:

3000 LET Y= f(X)

El programa contiene inicialmente en dicha línea el siguiente ejemplo:

3000 LET Y= 85 + 70*SIN (X/30)

que sirve para visualizar una senoidal.

Hecho esto se pulsa RUN 1000 con lo cual el aparato pasa a modo FAST y calcula durante algunos segundos colocando los códigos apropiados en la zona de archivo. Terminados los cálculos se entra automáticamente en la visualización de la gráfica. Para interrumpirla se pulsa BREAK, y se regresa al menú pulsando RUN.

La introducción de varias gráficas puede hacer que las nuevas se "coman" pequeños pedazos de las antiguas, debido a que el programa sólo usa un punto en negro por carácter.

La opción 3 se limita a dar una nueva visualización de las gráficas introducidas hasta el momento.

La opción 4 permite desplazar la gráfica a derecha o izquierda en la pantalla. Puede ser útil si se modifican las instrucciones de llamada al CM, lo cual desarregla o descentra la gráfica. El desplazamiento se obtiene introduciendo un número entre 0 y 100. El cen-

trado está inicialmente ajustado a 11. Un número mayor produce un desplazamiento a la izquierda y uno menor a la derecha.

La opción 5 permite usar un número de líneas distinto de 180. Al usar esta opción debe recordarse que las líneas primera y última se emplean para el trazado de los márgenes superior e inferior, por lo que no se puede dibujar en ellas, y que la modificación del número de líneas borra las gráficas previamente introducidas.

La zona útil de la pantalla corresponde a rangos de valores de las variables comprendidos entre 1 y 192 para la X y entre 1 y 178 para la Y.

Los valores de la X los asigna el programa mediante un ciclo FOR-NEXT. Los valores de la Y que salen fuera del rango simplemente no aparecen en pantalla, pero aquellos que no pueden ser calculados debido a que exigen operaciones aritméticas ilegales (por ejemplo, radicandos negativos en una raíz cuadrada, argumentos nulos o negativos en logaritmos, denominadores nulos, etc.) detienen el cómputo forzando un código de error. La opción 3 permite entonces visualizar el pedazo de gráfica obtenido hasta el momento.

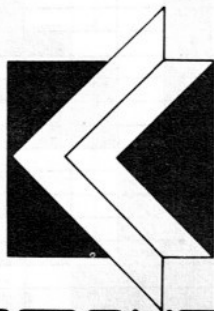
Los tramos de pendiente elevada pueden tener un aspecto algo discontinuo. Esto se puede arreglar con un paso (STEP) menor que 1 en la línea 2000, aunque a costa de prolongar el tiempo de cálculo.

Miguel A. Lerma.

PON «EN ORDEN» TU FUTURO

NUESTRAS MARCAS

COMMODORE-64 Y VIC-20
SINCLAIR (ZX Y SPECTRUM)
ORIC
ATARI
UNITRON
NEW BRAIN
TEXAS INSTRUMENTS



**COMPUTER
CENTER**

NUESTROS SERVICIOS

ASESORAMIENTO INFORMATICO
ESCUELA DEL ORDENADOR
CLUB DEL ORDENADOR
PERSONAL
CENTENARES DE PROGRAMAS:
EDUCATIVOS
RECREATIVOS
DE GESTION

PERIFERICOS:
IMPRESORAS
UNIDADES DISCO
MONITORES
CASSETTES
TELEVISION
SUMINISTROS

TODOS TIPO DE FACILIDADES DE PAGO
REMISION A TODA ESPAÑA

VISITANOS O PONTE EN CONTACTO

RBLA. DE CATALUNYA, 50, 1º 2ª. TELF. (93) 215 60 70
APART. CORREOS 5455 - 08007 BARCELONA

LABERINTO

Este mes os propongo un juego, al que he denominado LABERINTO, que puede proporcionaros $(A \cdot \sin(E) / \tan(z)) \cdot 5 \cdot 10^{999}$ minutos de entretenimiento a la par que, como cualquier programa, elementos para el estudio de la mecánica del lenguaje BASIC.

Está desarrollado con un microordenador SHARP MZ.80.B, Interpreter SB-6610, pero, tranquilos!, ya que leyendo el artículo que acompaña al programa estareis en condiciones de modificarlo para la mayoría de los micros que corren por aquí.

DESCRIPCION DEL JUEGO:

La mecánica del juego es muy sencilla. Se trata de encontrar el recorrido seguido por el ordenador, dentro de un tablero de 25 casillas numeradas. De forma aleatoria, como siempre!, se asignan un número de pasos al recorrido, una casilla de salida y otra de llegada; a continuación, empezando por el primer jugador (1 - 4 jugadores) se debe introducir los números de las casillas por las que se supone ha realizado el recorrido el ordenador. Siempre, como primer número del recorrido, se debe entrar el correspondiente a la casilla de salida. Al entrar los pasos sonará una música aguda indicando que el paso es correcto ó una música grave que nos indicará error y cederá el turno al siguiente jugador.

El recorrido, dentro de la misma partida, es idéntico para todos los jugadores, por lo que el camino seguido por un jugador hasta entrar un paso incorrecto puede servir para el jugador siguiente, que deberá intentar recordarlo, ya que tendrá que empezar de nuevo por la casilla de salida.

Si los pasos entrados han sido correctos al llegar a la última casilla sonará la música de fin de laberinto y se felicitará al jugador que lo haya conseguido. Se puede continuar jugando de forma que el marcador de errores no se restaure, o sea que se vayan acumulando, ó bien ejecutar de nuevo el programa, con lo que el marcador se restaura a cero. Atención, porque el recorrido del laberinto puede cambiar de sentido y volver sobre si mismo en cualquier punto, o incluso pasar sobre la casilla de salida o llegada varias veces.

Hay muchas formas de establecer el ganador en el juego del laberinto, puede jugarse a una partida, siendo ganador el primero que descubra el recorrido, ó bien a series de partidas, asignando puntos positivos por laberinto descubierto, dependiendo del número de pasos del recorrido y el número de

intentos realizados. Se puede jugar, también, en solitario, con la compañía del ordenador, por supuesto!.

DESCRIPCION DEL PROGRAMA:

No es fácil describir el programa ya que es sumamente sencillo.

El número de pasos del laberinto se asigna de forma aleatoria, teniendo como tope, el máximo entrado por el operador, en la rutina de entrada pasos laberinto, líneas 500-535. La variable NE contendrá el número de pasos asignado.

GENERACION DEL LABERINTO: (líneas 400-496) De forma aleatoria se asigna una casilla de salida (línea 405). A partir de la salida los siguientes pasos, hasta llegar a NE pasos, se realizan de la siguiente forma: Se genera una variable, D, aleatoria y comprendida

EJECUCION DEL PROGRAMA

00000000 LABERINTO 00000000

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

SALIDA	8
LLEGADA	3
LONGITUD	8
PARTIDA	

ERRORES

MIGUEL = 1
JOSE = 1

MIGUEL ENTRA TU TIRADA (1-25) = 17

entre 1-4, que indicará la dirección a seguir desde la última casilla encontrada, dependiendo de D la dirección será = 1: abajo, 2:arriba, 3:derecha y 4:izquierda. (ON D GOTO... línea 430). Se comprueba que exista una casilla en la dirección indicada por D antes de asignarla (líneas 420 a 425), ya que, por ejemplo, no existe una casilla arriba de la 1.

Los pasos se acumulan en la matriz numérica E(n). Si L, que indica el número de orden del paso, es igual al número de pasos del laberinto (línea 475) se asigna como casilla de llegada la correspondiente a este paso (ver línea 485).

— GOSUB 250: Hace que se visualicen en el teclado las marcas de salida y llegada, la tabulación para la impresión en la casilla correcta se calcula en 265-270.

— La rutina ENTRADA MOVIMIENTO, líneas 200-270, nos sirve para entrar las tiradas y comprobar si son correctas. J\$(n) = Contiene el nombre de los jugadores. En la línea 220 comprobamos que el número introducido esté dentro de los márgenes de la tabla y que no se repita el de la última introducción (U = último número entrado).

El valor de Z indica: Si es = 0 que el paso es erróneo, si el número entrado, Q, es igual a la casilla de llegada y el número de paso es igual al número de pasos del laberinto (NE): Z = 2, lo que indicará que hemos finalizado el recorrido. Caso contrario Z se igualará a 1, indicando paso correcto.

Como variables musicales he utilizado 4:

Z3\$ = Mi típica variable de error de operación (3 pitidos agudos).
M\$(Z) = dependiendo del valor de Z tenemos:

M\$(0) = Música de paso incorrecto (Pitido grave).

M\$(1) = Música de paso correcto (Varios tonos agudos).

M\$(2) = Música final laberinto (Música de tonos agudos).

Vistas las principales rutinas del programa veamos la conversión a otros ordenadores:

TEMPO 7: Indica la velocidad de ejecución de la música (No es importante).

TODAS LAS VARIABLES DE MUSICA (Z3\$, M\$, M\$(n)) pueden suplirse,

dependiendo del ordenador, por SOUND, BEEP, PLAY,... En el caso de no tener ordenador con 'cualidades musicales' puede suplirse las instrucciones MUSIC por un mensaje de 'PASO CORRECTO', 'FIN LABERINTO', etc. . .

CURSOR x,y = Sitúa el cursor en la columna 'x', fila 'y' de la pantalla. Se puede suplir por LOCATE x,y;PRINT TAB (x,y) ó PRINT@n (siendo n el número de la posición de pantalla).

GET X\$ = Acepta una pulsación del teclado. Es equivalente a X\$=INKEY\$, ó X\$=KIN\$.

CONSOLE C40,S0,24 = Es un comando de modo de visualización, compuesto, en este caso de 2 funciones = CONSOLE C40 = Pantalla en modo de 40 caracteres por línea, y CONSOLE S0,24 = que indica scroll automático desde la línea 0 hasta la 24. La primera función equivale a WIDTH 40, (ó 36 según pantalla), ó CTRL (n) (siendo n

```

1 REM -----
2 REM --- L A B E R I N T O ---
3 REM -----
4 REM -- AUTOR J.M. VIDAL LACASA --
5 REM -----
6 REM --- COPYRIGHT EL AUTOR Y ---
7 REM --- EL ORDENADOR PERSONAL ---
8 REM -----
9 REM
10 DIM E(100),J$(4),J(4),P(4),M$(3),C(200)
15 TEMPO 7:Z3$="+B1R0+B1R0+B1R0"
20 M$(0)="-C9":M$(1)="+G2R0F5R0G2"
25 M$="+F5R1+F5R1+B3R1+F5R1+F5"
30 CA$="00000000 L A B E R I N T O 00000000"
35 M$(2)=M$+"R5":M$:GOSUB 2000:GOSUB 300
40 PA=PA+1:GOSUB 500:GOSUB 400:T=1
45 U=0:N=1
50 GOSUB 200:MUSIC M$(Z)
55 ON Z GOTO 70,80
60 P(T)=P(T)+1:CURSOR 19,(T+12)
65 PRINT J$(T):T=T+1:IF (T=5) + (T>NJ) THEN T=1
66 GOTO 45
70 REM ----- PASO CORRECTO -----
75 N=N+1:GOTO 50
80 REM ----- FIN LABERINTO -----
85 CURSOR 0,20:PRINT "FELICIDADES ":J$(T):STRING$( " ",17)
90 CURSOR 10,21:PRINT "SEGUIR JUGANDO? (S/N)"
95 X$="":GETX$:IF X$=" " THEN 95
100 IF X$="S" THEN 115
105 IF X$="N" THEN PRINT CHR$(6):" A D I O S !!!":END
110 GOTO 95
115 CURSOR 10,21:PRINT STRING$( " ",29)
120 CURSOR 0,20:PRINT STRING$( " ",39)
125 FOR Y=3 TO 15 STEP 3:FOR X=3 TO 15 STEP 3
130 CURSOR X,Y:PRINT " "
135 NEXT X:NEXT Y
140 GOTO 40
200 REM ---- ENTRADA MOVIMIENTO ----
205 CURSOR 0,20:PRINT STRING$( " ",39)
210 CURSOR 0,20:PRINT J$(T):
215 INPUT "ENTRA TU TIRADA (1-25) = ":Q
220 IF (Q<1)+(Q>25)+(Q=U) THEN MUSIC Z3$:GOTO 205
225 U=Q:IF Q>E(N) THEN Z=0:RETURN
230 IF (Q=LL)*(N=NE) THEN Z=2:RETURN
235 Z=1:RETURN
250 REM --- MARCAS DE SAL/LLEG ---
255 N=S:V$="S":GOSUB 265
260 N=LL:V$="L":GOSUB 265:RETURN
265 A=INT(N/5+.9)
270 Y=A*3:X=(N-(A-1)*5)*3
275 CURSOR X,Y:PRINT V$:RETURN
300 REM ----- CONFECCION TABLA -----
305 PRINT CHR$(6):CA$
310 PRINT"
315 PRINT"
320 PRINT"
325 PRINT"
330 PRINT"
335 PRINT"
340 PRINT"
345 PRINT"
350 PRINT"
355 PRINT"
360 PRINT"
365 PRINT"
370 PRINT"
375 PRINT"

```

1	2	3	4	5	SALIDA
6	7	8	9	10	LONGITUD
11	12	13	14	15	PARTIDA
16	17	18	19	20	


```

380 PRINT" |21|22|23|24|25|
385 PRINT"
390 RETURN
400 REM ----- GENERAR LABERINTO -----
401 CURSOR 0,20:PRINT "UN MOMENTO, ESTOY CREANDO EL LABERINTO"
405 R=INT(RND(5)*25)+1:S=R:P=S:L=1
410 E(L)=P:C(L)=P
415 F=E(L):D=INT(RND(5)*4)+1
420 IF (D=3)*((P=5)+(P=10)+(P=15)+(P=20)) THEN 405
425 IF (D=4)*((P=6)+(P=11)+(P=16)+(P=21)) THEN 405
430 ON D GOTO 435,440,445,450
435 F=F+5:GOTO 455
440 F=F-5:GOTO 455
445 F=F+1:GOTO 455
450 F=F-1
455 IF (P<1)+(P>25) THEN 415
460 IF (L>1)*(P=E(L-1)) THEN 415
475 L=L+1:IF L=NE THEN 485
480 GOTO 410
485 E(L)=P:LL=P:GOSUB 250
490 CURSOR 29,3:PRINT "":CURSOR 29,5:PRINT " "
495 CURSOR 29,3:PRINT STR$(S):CURSOR 29,5:PRINT STR$(LL)
496 CURSOR 0,20:PRINT STRING$( " ",40):RETURN
500 REM ----- ENTRAR No. PASOS -----
505 CURSOR 0,20:INPUT "No. MAX. PASOS LABERINTO (10-100) = ";NE
510 IF (NE<10)+(NE>100) THEN MUSIC Z3$:GOTO 505
515 K=INT(RND(5)*NE+1):IF K<7 THEN 515
520 NE=K
525 CURSOR 0,20:PRINT STRING$( " ",39)
530 CURSOR 29,7:PRINT " "
535 CURSOR 29,7:PRINT STR$(NE):RETURN
2000 REM ----- ENTRADA JUGADORES -----
2005 CONSOLEC40,50,24:PRINT CA$
2010 CURSOR 0,10:PRINTSTRING$( " ",39)
2015 CURSOR 0,10:INPUT "No. DE JUGADORES (1-4) = ";J:NJ=J
2020 IF (J<1)+(J>4) THEN MUSIC Z3$:GOTO 2015
2025 FOR Q=1 TO J
2030 CURSOR 0,10:PRINT STRING$( " ",39)
2035 CURSOR 0,10:PRINT "NOMBRE JUGADOR ";Q;
2040 INPUT " = ";J$
2045 J$=LEFT$(J$+" ",10)
2050 J$(Q)=J$:NEXT Q:RETURN

```

el carácter correspondiente, que depende del ordenador). Esta instrucción se puede suprimir totalmente en los ordenadores que solo tengan un formato de visualización.

STRING\$("X", n) = (imprime una serie de n caracteres X. Atención a esta instrucción porque es muy similar en sintaxis a la de TRS, VIDEO GENIE ó TOSHIBA, pero dentro del paréntesis los datos están al revés, primero el carácter y después el número de elementos. Equivalentes: STRING\$(n, "X").

Por último aclarar como funcionan los operadores lógicos:

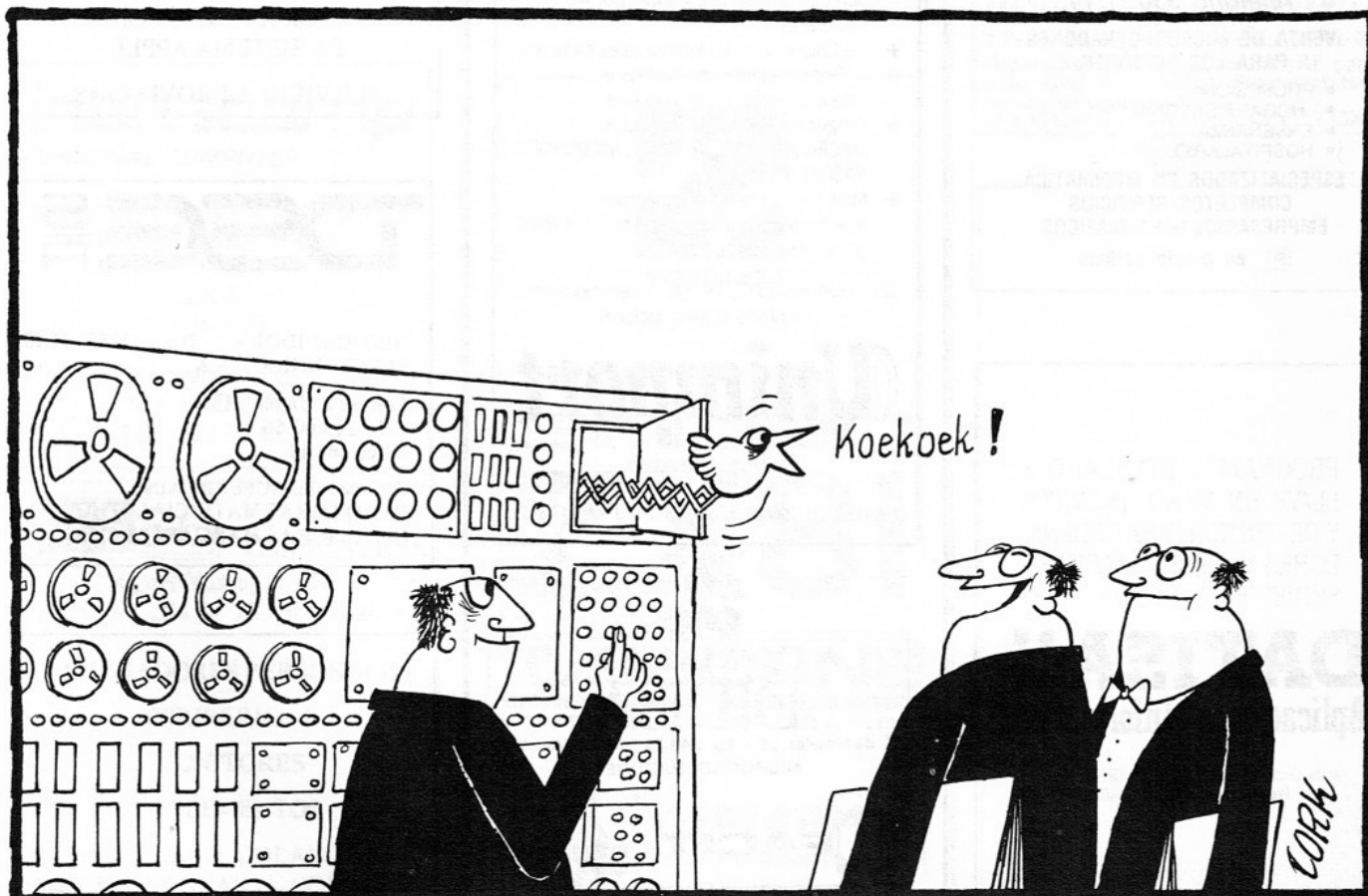
- IF <expresión> + <expresión> THEN <número línea>
- IF <expresión> * <expresión> THEN <número línea>

El símbolo '+' corresponde al operador lógico = OR.

El símbolo '*' corresponde al operador lógico = AND.

Para finalizar solo me queda desearos muy buenos ratos de ocio con 'LABERINTO' y como siempre esperar vuestras críticas y sugerencias. □

José M. Vidal Lacasa.



DIRECTORIO

EL ORDENADOR PERSONAL

1000 ordenadores. Material

ACCORD
microsistemas

Software
para aplicaciones
verticales.

DISTRIBUIDORES OFICIALES DE
COMMODORE y OLIVETTI M20.

Apartado de Correos 10.048, Madrid. Tel. (91) 448 38 00.



**DATA
PROCESSING 2000,
S. A.**

**EN MICROINFORMATICA,
INFORMESE ANTES**

**Sabino Arana, 22-24, bajos.
Barcelona-28.
Teléfono 330 77 14.**

**VENTA DE MICROORDENADORES
PARA LOS SECTORES:**

- PROFESIONAL.
- HOGAR/PERSONALES.
- ENSEÑANZA.
- HOSPITALARIO.

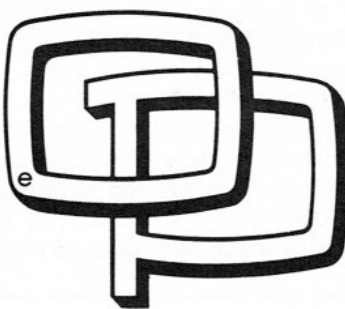
**ESPECIALIZADOS EN MEDIMATICA.
COMPLETOS SERVICIOS
EMPRESARIOS/INFORMATICOS.**

[P] en propio edificio.

PROGRAMAS STANDARD Y
LLAVE EN MANO, TECNICOS
Y DE GESTION PARA ORDENA-
DORES HEWLETT - PACKARD
SERIES 80, 9.800, 200 Y 250

DATISA
Aplicaciones Informáticas

Avda. Generalísimo, 25-1º B. Tel. (91) 715 92 68
Pozuelo de Alarcón. MADRID-25



**ATARI® 600XL
ATARI® 800XL**

**ORDENADORES
PARA EL HOGAR**

Extenso software listo para el uso

- ★ Microprocesador: 6502 (ciclo de 0,56 Microsegundos 1,8 MHz), ANTIC, GTIA, POKEY (espec.)
- ★ Gráficos de alta resolución (320.192) puntos. Pantalla de 24 líneas por 40 caracteres.
- ★ 16 Colores con 16 Intensidades cada uno.
- ★ 4 Sintetizadores simultáneos e independientes. Cuatro octavas.
- ★ Lenguajes: BASIC, ASSEMBLER, MACRO-ASSEMBLER, PILOT, MICROSOFT, PASCAL Y otros.
- ★ Módulos de memoria conectables directamente por el usuario de 16 K RAM, 32 K RAM y 64 K RAM.

Distribuidores EXCLUSIVOS y servicio técnico
en todo el área nacional.

Unimport

División Ordenadores
c/ Dos Amigos nº 3 Madrid-8
Apartado de Correos 8286 Tels. 247 31 21- 247 31 26



**Diez & Diez, S.A.
DIDISA**

Pº. de Rosales, 26 • Tls. 248 24 01-02 • Madrid-8
MICROORDENADORES



FACIT



Conde de Borrell, 108
Tel.: 254 45 30
BARCELONA 15

Micro Ordenadores:
Rockwell
Ohio Scientific
Videogenie
Sinclair



- MICROTERSA

Miguel Yuste, 16-2ºB
Teléfono: 754 04 73 - MADRID-17

**ORDENADORES PERSONALES
UNA EMPRESA CON VOLUNTAD
DE SERVICIO**

**MICROPROCESADORES
COMPATIBLES CON
EL SISTEMA APPLE**

SERVICIO A PROVINCIAS



DISTRIBUIDORA DE SISTEMAS
ELECTRONICOS, S.A.

Comtes d'Urgell, 118
Tel.: 323 00 66
Barcelona 11

Ordenadores **SUPERBRAIN**
IMPRESORAS MATRICIAL ITHO
IMPRESORAS MARGARITA ITHO



Sandoval, 4
Tel.: 445 18 33 - 445 18 70
MADRID - 10

Micro Ordenadores:
Rockwell
Ohio Scientific
Videogenie
Sinclair



DISTRIBUIDORES
AUTORIZADOS DE:

digital

**hp HEWLETT
PACKARD**

RANK XEROX

Su problema específico,
tiene
una solución específica.

IBERICA DIGITAL, S.A.

Informática profesional y de gestión.
CLARA DEL REY, 55 - MADRID - 2
TEL: 413 06 11.

indescorp
PERSONAL COMPUTER

ESPECIALISTAS EN SOFTWARE
(PROGRAMAS) PARA:

ZX-81
VIC - 20

Pº de la Castellana, 179 - 1º izq.
MADRID- 16
Tel.: 279 31 05

IEESA

- MICROTHERSA

Miguel Yuste, 16-2ºB.
Teléfono: 254 04 73 - MADRID-17

ORDENADORES PERSONALES

PERIFERICOS

MONITORES

SUPLEMENTOS

SE PRECISAN COLABORADO-
RES DE VENTAS A COMISION.



INVESTRONICA

Tomás Breton, 21
Tel.: 468 01 00
MADRID 7

sinclair
ZX81

OSBORNE
COMPUTER CORPORATION

Cromemco
Incorporated
Tomorrow's Computers Today

LOGIMATICA

CONCESIONARIO AUTORIZADO
DEL ORDENADOR PERSONAL IBM.

Conoce los nuevos precios
del PC-IBM y sobre todo
sus nuevos programas?

En cualquier caso le aseguramos un
estudio serio y profesional de sus
necesidades, ofreciéndole:

- Software específico "llave en mano"
- Experiencia en comunicaciones.
- Cursos de formación de usuarios.
- Aplicaciones sectoriales:
- Software standar de aplicación y gestión:

• Paquetes inte-
grados para
profesionales
• Finanzas
• Contabilidad
• Hostelería
• Educación
• Agentes de Se-
ruidos

• Contabilidad
• Almacenes
• Facturación
• Nomina
• Tratamiento
de datos
• Hojas electrónicas
• Bases de Datos
• Tesorerías

LAGASCA, 90
(esquina Ortega y Gasset)
Madrid-6
Telf.: 431 60 32
435 52 56



**MECOMATIC
SHARP**

MECANIZACION DE OFICINAS, S. A.

BARCELONA-36
Av.Diagonal, 431 bis. Tfno.200 19 22
MADRID-3
Sta.Engracia, 104 Tfno.441 32 11
BILBAO-12
Iparraguirre, 64 Tfno. 432 00 88
VALENCIA-5
Ciscar, 45 Tfno. 333 55 28
SEVILLA-1
San Eloy, 56 Tfno. 215 08 85
ZARAGOZA-6
J.Pablo Bonet, 23 Tfno. 27 41 99
Ordenadores profesionales SHARP para
todo nivel de actividad. Programas tec-
nicos y de gestión.
SERVICIO TECNICO GARANTIZADO

Consulte sobre nuestros cursos de BASIC y Pascal para estudiantes de BUP - COU - Escuelas Técnicas - Universitarios - Profesionales - Empresas y adultos en general.

Por vez primera en España cursos de iniciación y tarifas especiales para amas de casa y para la tercera edad.

MICRO SPOT

El centro MICRO SPOT, especializado en informática, que ofrece la oferta más amplia en microordenadores y una variada gama de periféricos, impresoras, unidades de cassette y disquette, monitores color y F.V., etc. Disponemos de completos listados de software en cinta y disco, para programas técnicos, de aplicación, educativos y juegos. Accesorios diversos, manuales, libros técnicos y revistas especializadas.

Conde de Cartagena, 9 (zona Retiro) - Madrid-7 - Tels. 251 32 04/05/06/07



Programas específicos para arquitectura, construcción y obra civil, sobre microordenadores Hewlett-Packard. Pídanos Catálogo gratuito.

SOFT biblioteca de programas

Apertado de Correos, 10.048. Tel. (91) 448 35 40. Madrid.

MAYBE

General Martínez Campos, 5 Bajo Izqda.
Tel.: 446 60 18
MADRID - 10
Brusi, 102 - Entresuelo 3º.
Tel.: (93) 201 21 03.
BARCELONA - 6

Distribuidores de los ordenadores: Apple II y Apple III y de los discos rígidos COVRVUS de 5, 10 y 20 Megabytes.



Sistemas y Servicios

La única Tienda de Ordenadores especializada en la mecanización de la Pequeña y Mediana Empresa donde en cualquier momento podrá discutir:

- Análisis Mecanización de su Empresa.
- Desarrollo de Programas a Medida

HEWLETT PACKARD - HP 150
WANG PC
TOSHIBA T300, T100
VICTOR/SIRIUS

Numerosas instalaciones. Empresas nos avalan.

Venta en Provisión. Zona Centro
Servicio Técnico Propio

Juan Alvarez Mendizabal, 55. MADRID-8
(En Arguelles, antes Victor Pradera)
Teléfonos: (91) 242 15 57 y 67.

LOGIMATICA

en
Lagasca, 90
(esquina Ortega y Gasset)
MADRID-6

UN NUEVO CONCESIONARIO
DE INVESTRONICA PARA
EL ORDENADOR SINCLAIR

SINCLAIR ZX 81 14.975 Ptas.
SINCLAIR ZX SPECTRUM 16 k 39.900 Ptas.
SINCLAIR ZX SPECTRUM 48 k 52.000 Ptas.

A UN SIN FIN DE PROGRAMAS PARA
JUEGOS, EDUCACION Y UTILIDADES
ECONOMICAS Y GESTION.

NO PERDA EL TRINDE LA INFORMATICA

Visitenos portando
este anuncio y ob-
tendrá condiciones
especiales

Hnos: 431 60 32
435 52 56
TELSPERAMOS



7000 Sistemas en Kit



ELECTRONICA

SANDOVAL S.A.

COMPONENTES ELECTRONICOS PROFESIONALES
TEL. (91) 445 18 33 - 445 18 70
MADRID - 10

Sandoval, 4
Tel.: 445 18 33 - 445 18 70
MADRID - 10

Micro Ordenadores:
Rockwell
Ohio Scientific
Videogenie
Sinclair

8000 Libros y Revistas

PRODACE

Ferraz, 11 - 3º
Tel.: 247 30 00
MADRID 8

Programación de Ordenadores en Basic.

Tiendas de Informática.

MPF-I/ IP/65

CPU Z80/6502

EL SISTEMA
IDEAL PARA
APRENDER EL
FUNCIONAMIENTO
DE UN
MICROPROCESADOR

Aplicaciones industriales - OEM
Centros de Formación Profesional



MPF-II-64K

PEQUEÑO
COMPATIBLE
POTENTE

CARACTERISTICAS TECNICAS

CPU 6502 - ROM 16KB - RAM 64KB
INTERPRETE BASIC COMPATIBLE APPLESOF
FORMATO DE PANTALLA

- 960 caracteres (24 líneas, 40 columnas)

GRAFICOS

- (Baja resolución) en matriz de 40 x 48
- (Alta resolución) en matriz de 280 x 192

TECLADOS

- 49 Teclas alfanuméricas y de funciones

INTERFACE DE CASSETTE

INTERFACE DE CARTUCHO-SOFTWARE

INTERFACE IMPRESORA CENTRONICS

SALIDA DE VIDEO

- Conectable a monitor monocromo (Video comp.)
así como TV color PAL

ALTA VOZ INCLUIDO



AMSTRAD

LO INCREIBLE

El Basic del Amstrad es rápido, más rápido que casi todos los Basic de 8 bits y que algunos Basic de 16 bits.
PERSONAL COMPUTER WORLD MAYO 84

Amstrad, con su nuevo CPC-464, ha demostrado ser un campeón en saltos de longitud.
COMPUTER ANSWER AGOSTO 84

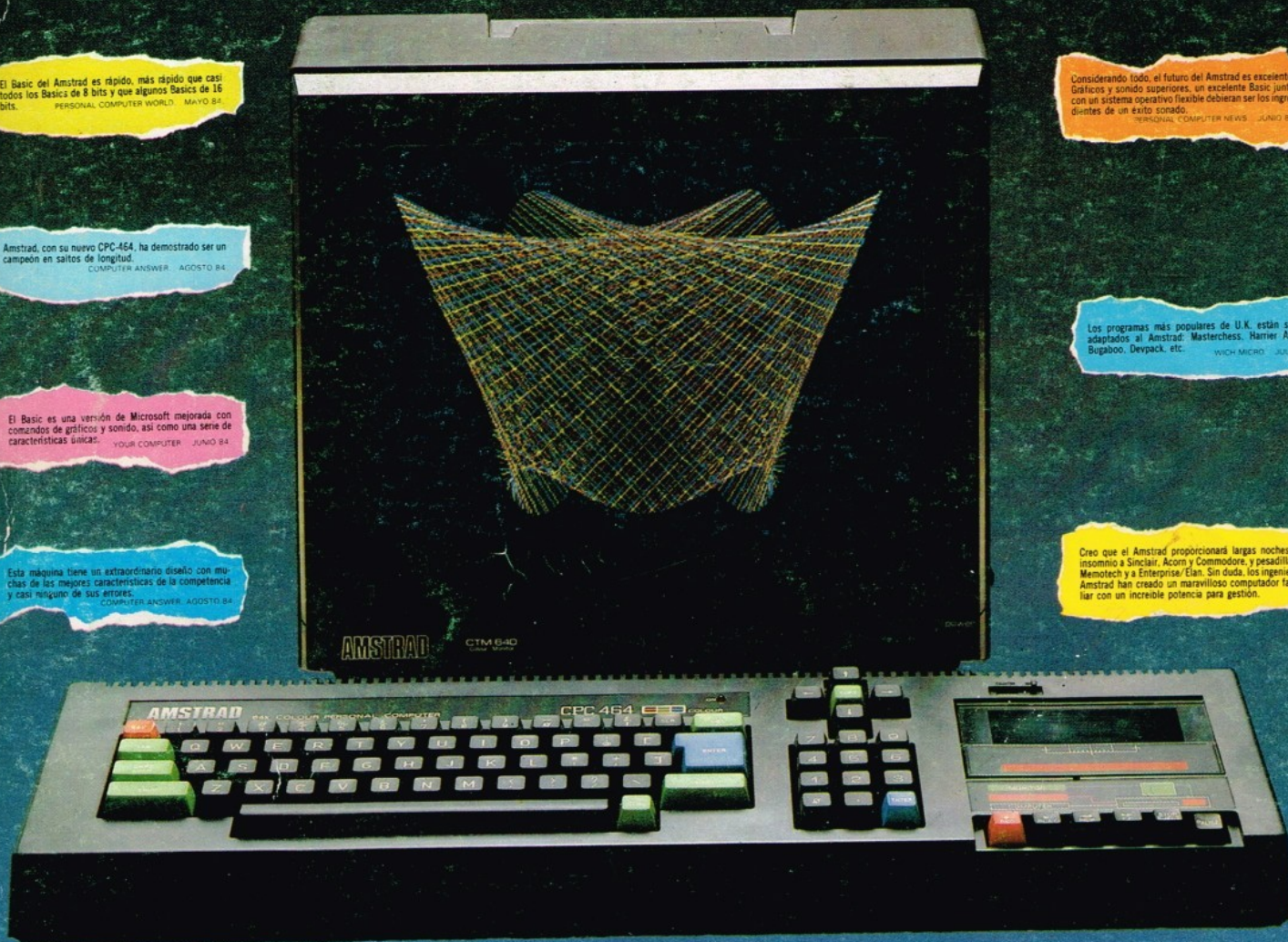
El Basic es una versión de Microsoft mejorada con comandos de gráficos y sonido, así como una serie de características únicas.
YOUR COMPUTER JUNIO 84

Esta máquina tiene un extraordinario diseño con muchas de las mejores características de la competencia y casi ninguno de sus errores.
COMPUTER ANSWER AGOSTO 84

Considerando todo, el futuro del Amstrad es excelente. Gráficos y sonido superiores, un excelente Basic junto con un sistema operativo flexible deberían ser los ingredientes de un éxito sonado.
PERSONAL COMPUTER NEWS JUNIO 84

Los programas más populares de U.K. están siendo adaptados al Amstrad: Masterchess, Harrier Attack, Bugaboo, Devpack, etc.
WICH MICRO JUNIO 84

Creo que el Amstrad proporcionará largas noches de insomnio a Sinclair, Acorn y Commodore, y pesadillas a Memotech y a Enterprise/Elan. Sin duda, los ingenieros Amstrad han creado un maravilloso computador familiar con un increíble potencia para gestión.



UNIDAD CENTRAL CON 64 K. MAGNETOFONO Y MONITOR EN COLOR

UNIDAD CENTRAL CON 64 K. MAGNETOFONO Y MONITOR EN FOSFORO VERDE P.V.P. 89.900 pts

- Monitor en color o en fósforo verde incluido en el sistema.
- Magnetófono incorporado de alta velocidad (1.000 ó 2.000 baudios).
- Memoria standard de 64 K de RAM ampliables hasta 8.160 K y 32 K de ROM.
- Gráficos en alta resolución de hasta 640x200 pixeles direccionables individualmente.
- Texto en pantalla de 20, 40 y 80 columnas por 25 líneas mediante un sencillo comando Basic.
- Tres canales de sonido con siete octavas y salida stereo.
- Basic extendido con funciones de Edición: Delete, Renumber, Auto, Trace, De lenguaje estructurado: If, Then, Else, While, Wend, De control de Procesador: Every, After, De alta resolución: Plot, Draw, etc.
- Tres modos de pantalla con una paleta de 27 colores y efectos de "flash".
- Microprocesador Z80 (4 Mhz) con implementación de CP/M.
- 74 teclas profesionales tipo "Qwerty" con bloque numérico y teclas para cursores.
- Lector de discos de 3" y 170 K. Opcional con CP/M y LOGO incluidos en el sistema.
- Completo set de caracteres de 8 bits definibles por el usuario.
- 32 teclas programables con cadenas de 32 caracteres.

- Ocho ventajas de trabajo definibles por el usuario en la pantalla del monitor.
- Bus de Entrada/Salida para conexión a lectores de Discos, Modems y todo tipo de comunicaciones.
- Port para impresora Paralelo Centronics.
- Posibilidad de direccionamiento y utilización de hasta 240 bloques de 16 K ROM.
- Modulador opcional para utilización de T.V. doméstica.
- Extenso soporte de Software con más de 100 títulos ya disponibles entre juegos, educativos, programas profesionales y lenguajes (Ensamblador, Pascal, etc.).
- Manual del Usuario, de referencia Basic del Programador, de Firmware y tutorial traducidos al castellano.

Para mayor información:

AMSTRAD
Castellana, 179.
Tel. 270 43 28
28016 MADRID

Nombre _____
Dirección _____
Tel. _____